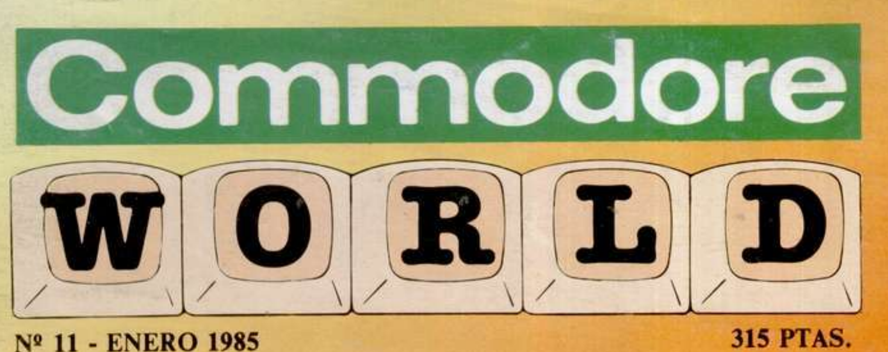


Nº 11 - ENERO 1985

315 PTAS.

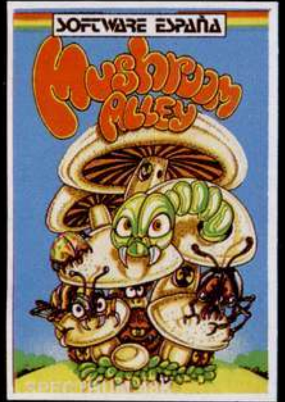
... y todas nuestras secciones
MAGIA - COLABORACIONES

Código máquina "a gogó"

Descubrir el secreto de la estructura interior de los Commodore
El arte de realizar organigramas
Castellanizar el Commodore-64
Rutinas Basic-3 (Paginación)
Constructor de datos
Utilidades y trucos
El duelo

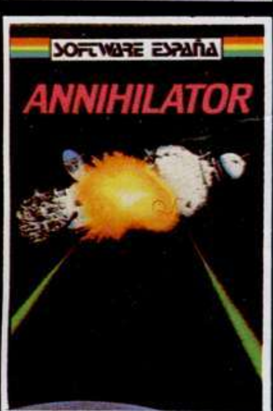
SOFTWARE ESPAÑA

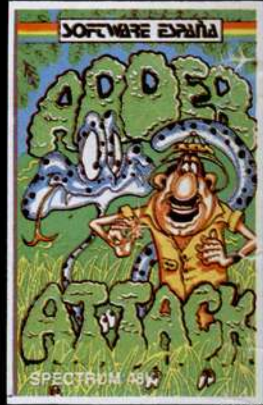
SOFTWARE ESPAÑA
Avenida de Arteijo, 19
15004 La Coruña
Teléf. (981) 25 51 72
Télex 47206 PPLL-E

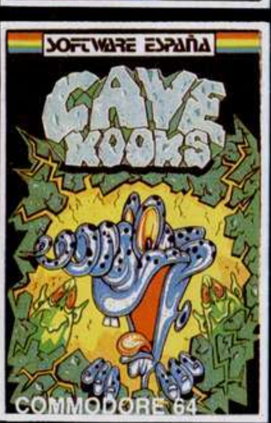


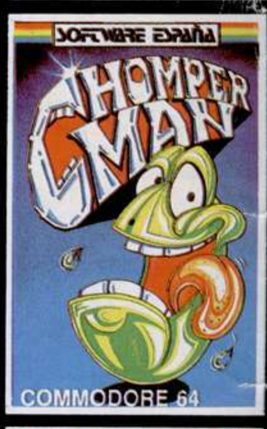


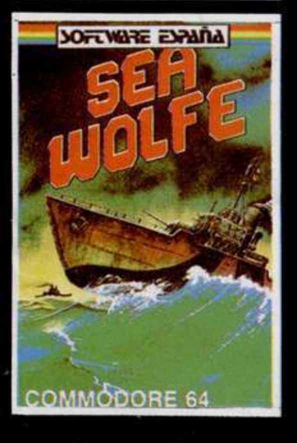


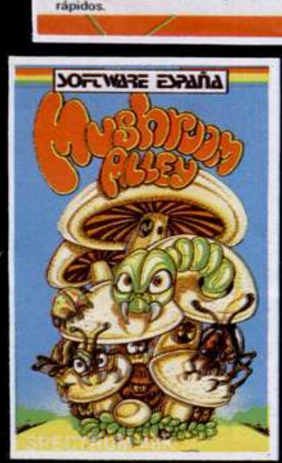


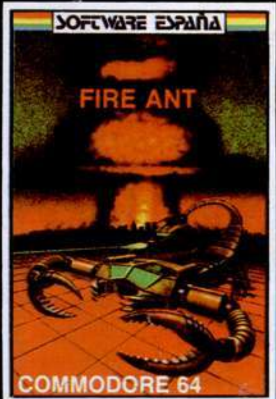


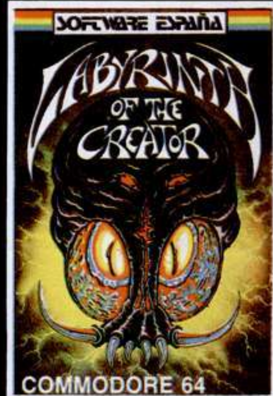


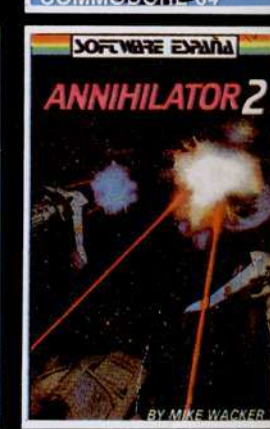












Commodore WORLD

Commodore World
está publicado por SIMSA
y la colaboración
de todos nuestros lectores

EQUIPO:
Manuel AMADO;
Juan L. AMESTI; Nieves CHESA;
José Luis ERRAZQUIN;
M.ª Jesús GARCIA;
Miguel A. HERMOSELL;
Alvaro IBAÑEZ; María LOPEZ;
Juan MARQUEZ; Juan MARTINEZ;
Pere MASATS; Victoria MORALES;
Rafael PARDO; Diego ROMERO;
Albert SANGLAS; Jordi SASTRE;
Valerie SHANKS; Francisco ZABALA...
...Y NUESTROS LECTORES

SIMSA
Coordinador María López
c/Barquillo, 21-3º Izda.
28004 Madrid. Teléf.: (91) 231 23 88/95
DELEGACION EN BARCELONA:
Sant Gervasi de Cassoles, 39 despacho 4.
08022 Barcelona
Tels.: (93) 212 73 45/212 88 48

Colabora MEC-COMMODORE
con Club Commodore
Coordinador Pere Masats
Valencia, 49-53 - 08015 Barcelona
Teléf.: (93) 325 50 08



Imprime:
IBERDOS, S. A.
Germán Pérez Carrasco, 24.
28027 Madrid
Depósito Legal: M-2944-1984

SUMARIO

CONTENIDO — PAG.

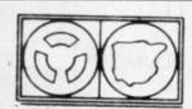






PROXIMO NUMERO

ESPECIAL IMPRESORAS
Cómo utilizar Centronics para los C-64 y los VIC.
Dibujo de una senoide con una impresora de margarita.
Caracteres españoles en impresoras OKI.
Dos grandes juegos: Cluedo y Cuevas de Alfa-ceti.
Programa de estadística... con deporte!!
Sumando con los "peques".
y todas vuestras colaboraciones.



Commodore World es miembro de CW Communications/Inc., el grupo editorial más grande y de mayor prestigio en el mundo en lo que se refiere al ámbito informático. Dicho grupo tiene a su cargo la edición de 52 publicaciones relacionadas con los ordenadores en 19 países. El grupo editorial está integrado por:
ALEMANIA: Computerwoche, Micro Computerwelt, PC Welt, Software Markt, CW Edition/Seminar, Computer Business, Run (Commodore). **ARABIA SAUDI:** Saudi Computerworld. **ARGENTINA:** Computerworld/Argentina. **AUSTRALIA:** Australasian Computerworld, Micro Computer Magazine, PC World. **BRASIL:** DataNews, Micro-Mundo. **DINAMARCA:** Computerworld/Danmark, Micro Verden. **ESPAÑA:** Computerworld/España. MicroSistemas, Commodore World. **ESTADOS UNIDOS:** Computerworld, Hot CoCo, InCider, InfoWorld, Microcomputing, PC World, 80-Micro, Mac World Micro Market Word, PC Jr. World Run (Commodore). **FINLANDIA:** Mikro. **FRANCIA:** Le Monde Informatique. **HOLANDA:** Computerworld Benelux, Micro/Info. **ITALIA:** Computerworld Italia. **JAPON:** Computerworld Japan, Perso Comworld, PC World. **MEXICO:** Computerworld/México, Compumundo. **NORUEGA:** Computerworld Norge, Mikro Data. **REINO UNIDO DE GRAN BRETAÑA:** Computer Management. Computer Business Europe. **REPUBLICA POPULAR DE CHINA:** China Computerworld. **SINGAPUR:** Asian Computerworld. **SUECIA:** ComputerSweden. MikroDatorn, Min Hemadator.

SUPER-INTERESANTISIMO

NUMERO EXTRAORDINARIO DE PROGRAMAS 2.ª EDICION

Nuestro numero Extra de Programas ha tenido un éxito monstruoso y sin apenas haber llegado a la calle se ha agotado. Se está preparando una segunda edición ya que las peticiones del mismo nos siguen llegando masivamente y los kioskos todavía no se han casi estrenado. Incluimos al pie de esta página un boletín de reserva para esta segunda edición, ¡¡CUIDADO!! los que ya hayáis enviado el boletín anterior, no es necesario que repitáis el envío, dentro de unos días tendréis el ejemplar en vuestra casa.

FORMA DE PAGO DEL EXTRAORDINARIO CON BOLETIN DE RESERVA

Solamente se aceptan peticiones que vengan acompañadas de cheque o giro. No se aceptan reembolsos.

BIENVENIDA A LA RADIOAFICION AL MUNDO COMMODORIANO Y DESPEDIDA AL MISTERIO DE LOS POKES

La serie el misterio de los pokes está tocando a su fin como ya habréis podido ver. Para ocupar el espacio que dejará, ER-4-ALI (que no es otro que Diego), ha preparado una serie de tres capítulos destinada a las aplicaciones del VIC-20 y el C-64 en el campo de la radioafición. Los que sean radioaficionados ya sabrán que su equipo se puede utilizar para recibir mensajes, y por supuesto, para enviarlos al éter. Se puede trabajar con el ordenador personal en las modalidades de CW y RTTY (telegrafía y radio-teletipo), ampliando con ello el campo de aplicaciones del ordenador. También se puede utilizar para calcular el rumbo ortodómico o trayectoria más corta entre dos puntos, orientación y diseño de antenas, etc. Los que no sean radioaficionados, quizás les despierta el gusanillo y comienzan a interesarse un poco por ese mundo tan curioso de las comunicaciones por radio, vía satélite, etc.

El motivo de escoger este tema ha sido la cantidad de cartas recibidas en la redacción solicitándolo. Para dar una idea de la difusión de esta afición en nuestro país, diremos que el número de aficionados con licencia oficial del ministerio pasa en estos momentos de trenta mil, y aproximadamente un cincuenta por ciento de ellos tienen también ordenadores personales.

Esperamos que la serie sea de interés para muchos de nuestros lectores.

RAMI Y ROMO AGREDIDOS POR EL CURSILLO DE CODIGO MAQUINA

Como veréis, en este número, por razones de espacio nos vemos obligados a quitar el comic que habitualmente aparece en nuestras páginas dirigido a los niños de dos a noventa años. La culpa de este desaguisado la ha tenido Diego, que nos ha invadido doce páginas de la revista con su cursillo de lenguaje máquina. Esperamos que los asiduos de esta sección no le esperen en la calle un dia de estos.

En el próximo número volverán nuestros pequeños amigos con sus aventuras, y ya hemos dado una buena reprimenda al autor de esta "fechoría" para que no vuelva a "pasarse de páginas".

IMPRESORAS PARA TODOS LOS COMMODORES

El próximo número va dedicado particularmente a las aplicaciones de impresoras.

En realidad lo que pasa es que aparecerán dos artículos dedicados a la utilización de impresoras que están dotadas de interface paralelo Centronics con los equipos Commodore (el VIC-20 y el C-64).

Como ya sabréis muchos, el tipo de interface Centronics es el que utilizan la mayoría de las impresoras del mercado, siendo por lo general el más barato dentro de las posibles opciones de conexión de cada modelo de impresora (RS-232, Centronics, etc.) por lo que se ha convertido en uno de los más utilizados por impresoras (si no es el más utilizado).

Con los dos programas que aparecerán podréis conectar otras impresoras a vuestros equipos sin la limitación de la gama de que dispone la casa Commodore. Además Diego comentará en la sección del cursillo de lenguaje máquina cómo poder modificar el programa (que por supuesto está escrito en código máquina), para poder adaptarlo a las peculiaridades de cada impresora (line feed automatico, etc.)... Pero ¡tranquilos!, los programas aparecen listados también con un cargador en basic para que cualquiera pueda teclearlos sin necesidad de saber código máquina, además; son muy cortos (no como la "cosa" que aparece en este número).

SORTEO DE DICIEMBRE DE COLABORADORES —PROGRAMAS Y MAGIA—

El sorteo de premios a los colaboradores cuyos esfuerzos han aparecido en los meses de julio a diciembre incluidos tendrá lugar, como dijimos, el 14 de diciembre en nuestra redacción. El único problema es que, debido a las fiestas de Navidad, este número lo cerramos antes de lo normal, hoy día 12, por lo cual no da tiempo a incluir los nombres de los ganadores en el mismo. La lista de los afortunados aparecerá en nuestro próximo número 12 del mes de febrero.

BOLETIN COMPRA DEL NUMERO EXTRAORDINARIO

2ª EDICION

NOMBRE ....

DIRECCION .... TELEF. ....

POBLACION .... C. P. .... PROVINCIA ....

DESEO .... EJEMPLAR (ES)

- PRECIO DE SUSCRIPCION 625 PESETAS ☐ • LECTOR NO SUSCRIPTOR (25% dtº) 935 PESETAS ☐
- NO SOY SUSCRIPTOR PERO ADJUNTO BOLETIN DE SUSCRIPCION Y ME ACOJO AL PRECIO ESPECIAL PARA SUSCRIPTORES DE 625 PESETAS ☐

- Adjunto cheque por .... Ptas. • Envío giro nº .... por .... Ptas.

Nº SUSCRIPTOR .... (este número va en todas las etiquetas con la que se os envía la revista)

Convocatoria del primer concurso nacional de programación en 80 columnas de Commodore-64 y 40-80 columnas del VIC-20 patrocinado por Ferre-Moret, S.A. Con la colaboración de Commodore- World, más de 500.000 ptas. en premios.

BASES:

1º Podrán participar todos los usuarios de COMMODORE-64 y VIC-20 que hayan obtenido la tarjeta de 80 columnas de FERRE-MORET, S.A.

2º Cada concursante podrá presentar, un programa realizado por él o por un equipo del que deberán indicar un titular.

3º Se enviará la explicación detallada del programa redactado en español, escrito a máquina. El programa se enviará en cassette o disco, adjuntando su correspondiente listado del programa.

4º Se enviarán por correo certificado a la siguiente dirección, FERRE-MORET, S.A. calle Buenos Aires, nº 30, 2º 3.ª de BARCELONA. Antes del 31 de enero de 1985.

5º Los programas no premiados podrán ser solicitados por sus propietarios en el plazo de treinta días después del fallo.

6º El resultado se comunicará, directamente a los ganadores y se publicará en la revista COMMODORE WORLD.

7º Los premios se entregarán durante la primera semana de marzo de 1985 coincidiendo con la entrega de premios del PRIMER CONCURSO NACIONAL ESCOLAR DE PROGRAMACION.

8º Los programas premiados quedarán en propiedad de FERRE-MORET, S.A.

9º Los programas que no obtengan premio, recibirán, en caso que así lo considere FERRE-MORET, S.A. ofertas de compra de sus programas.

10º Los premios ascenderán a 500.000 ptas. repartidos de la siguiente manera:

1º 250.000 Ptas.
2º 150.000 Ptas.
3º 100.000 Ptas.

11º Todos los participantes, que no hayan recibido premio u oferta de compra, podrán participar en el concurso de colaboradores que habitualmente COMMODORE WORLD viene realizando con un premio extra de FERRE-MORET, S.A. de 50.000 ptas.

12º Los trabajos deberán de ser inéditos y desarrollar temas de utilidad, programas profesionales, científicos, educativos, etcétera.

FERRE-MORET, S.A. se reserva el derecho de modificar las bases de este certamen, dejarlo desierto o suprimirlo, por causa de fuerza mayor. En este caso la empresa se compromete a comunicarlo a través de COMMODORE WORLD.

CONCURSOS PERMANENTES

Colaboraciones de Programas y Magia

La verdad sea dicha: os estáis portando; las colaboraciones que nos están llegando son numerosas y de muy buena calidad. Por lo tanto, vamos a aclarar las condiciones de nuestros concursos permanentes.

1) Los sorteos se realizarán en los meses de junio y diciembre.

2) Se sortearán siempre seis premios, tres de 15.000, 10.000 y 5.000 pesetas en metálico de "Commodore World" y tres del mismo valor en material de Microelectrónica y Control, entre todas las colaboraciones publicadas.

3) Se sortearán siempre seis premios, tres de 3.000, 2.000 y 1.000 pesetas en metálico de "Commodore World" y tres del mismo valor en material de Microelectrónica y Control, entre todas las contribuciones publicadas en la sección de "Magia".

4) Adicionalmente, se sortearán diversos premios de cintas, juegos, etc. donados por firmas y distribuciones, entre todas las colaboraciones.

5) Los autores de las colaboraciones vendidas dentro de nuestro "Servicio de Cintas" percibirán el 20% del precio de la cinta.

6) Todas las colaboraciones deben venir escritas a máquina y los programas grabados en cinta (si es posible), o con el listado completo en impresora. Nuestros lectores más jóvenes pueden escribir a mano pero con letra muy clara.

7) Quedan automáticamente descalificadas tanto del sorteo como del "Servicio de Cintas" las colaboraciones que hayan sido enviadas a otras revistas.

8) Las colaboraciones se enviarán a "Commodore World", c/Barquillo, 21-3º izda. 28004 Madrid.

MAS VIAJES Y OTRO REGALO DE "IMPRESION"... A POR EL SUSCRIPTOR 8.000

Es evidente que a nuestros suscriptores les gusta viajar. De momento Joan Pujiula visitará los Estados Unidos en julio y Juan Mejuto Iglesias irá a Inglaterra en el otoño. Queríamos sortear algo un poco especial ENTRE TODOS LOS SUSCRIPTORES cuando lleguemos al suscriptor número 8.000, y hemos optado por dos premios, uno concedido por Microelectrónica y Control y otro concedido por "Commodore World".

MEC regalará una impresora MPS 801 (o similar) y nosotros hemos escogido un surtido de viajes muy "chulos" para dos personas para que el ganador elija el que más le apetezca. La selección que hemos hecho es la siguiente:

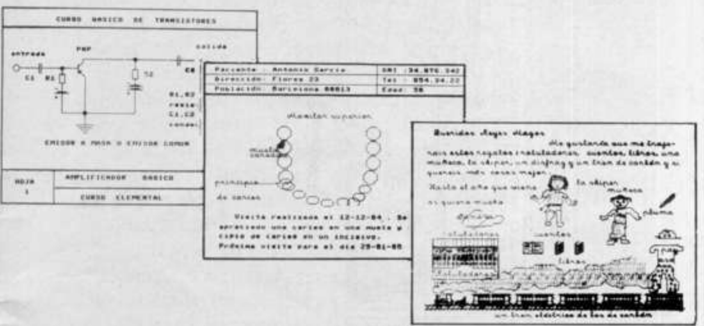
Viajes de 4 días en hoteles de primera clase a uno de estos lugares: París, Lisboa, Atenas, Amsterdam, Niza, Viena.
O viajes de una semana en hoteles de 4/5 estrellas: Canarias o Mallorca.

INDICE DE ANUNCIANTES



Entra con nuestros programas en el mundo de la Musica y el Diseño

■ LA OTRA MANERA DE ESCRIBIR



Escriba y dibuje sobre la misma hoja. Las posibilidades son ilimitadas... Informes, esquemas, ficheros gráficos, croquis con acotaciones, presupuestos, pequeños planos...

CARACTERISTICAS:
- Hoja de 50×40 caracteres (400×320 pixels)
- Modo texto (puede crearse su propio alfabeto)
- Modo gráfico mediante joystick (4 velocidades)
- Modo robot. Volcado de gráficos predefinidos (32×24 pixels) sobre la hoja (128 gráficos por fichero).
- Grabación de la hoja en diskette o cassette
- Salida por impresora en alta resolución
- Versión cartucho

SANDRA

■ EL INCREIBLE TECLADO MUSICAL

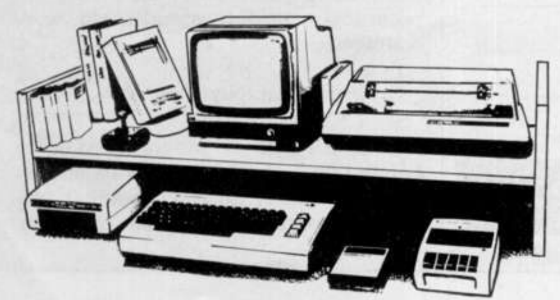


Transfórmese desde ahora en músico... creando, editando, componiendo y ejecutando sus propias piezas. El increíble teclado musical hace mucho más qe convertir su commodore en un sintetizador.

ARTICULO	VERSION	REFERENCIA
– INCREIBLE MUSICAL KEYBOARD	DISCO	C64D100
– INCREDIBLE MUSICAL KEYBOARD	CASSETTE	C64T100
– MUSIC PROCESSOR	DISCO	C64D101
– MUSIC PROCESSOR	CASSETTE	C64T101
– KAWASAKI SYNTHESIZER	DISCO	C64D102
– KAWASAKI SYNTHESIZER	CASSETTE	C64T102
– KAWASAKI RHYTHM ROCKER	DISCO	C64D103
– KAWASAKI RHYTHM ROCKER	CASSETTE	C64T103
– 3001 SOUND ODYSSEY	DISCO	C64D104
– 3001 SOUND ODYSSEY	CASSETTE	C64T104

SIGHT&SOUND Music Software, Inc

ORDENA TU ORDENADOR



Mueble sobremesa auxiliar

DESEO RECIBIR INFORMACION:

Nombre y dirección:



Por Robert BAKER (RUN EE.UU.)
Traducido por Valerie SHANKS y adaptado por Diego ROMERO.

Constructor de datos

C-64 y VIC-20

Esta utilidad fue diseñada para los sistemas PET de Commodore pero el Listado 1 está adaptado para funcionar con el C-64 y el VIC-20.

El Programa lee un programa en lenguaje maquina de disco y crea un programa en Basic en disco con el mismo programa en lenguaje máquina convertido en sentencias Data. El programa también añade un bucle For...Next con los parámetros correctos para leer los datos e introducirlos en la memoria mediante unos Pokes. Esto te proporciona un método sencillo para convertir programas en lenguaje máquina en una forma que puedan ser mezclados con un programa en Basic.

Cuando se ejecuta el programa lo primero que hace es pedirte el nombre del fichero del programa en lenguaje máquina que será leído y convertido (líneas 210-260). El programa en lenguaje máquina tiene que ser un fichero estandard y cargable como un programa (tipo PRG). No puede ser ningún tipo de fichero intermediario creado mediante un ensamblador. Si utilizas un ensamblador para crear el programa en lenguaje máquina, lo puedes cargar mediante el procedimiento normal y luego usar un monitor para salvarlo en disco como un fichero de programa. Una vez cargado como un fichero de programa, puede ser utilizado normalmente.

Lo Que Hace

Una vez que el programa sabe el nombre de fichero del programa en lenguaje máquina, pide el nombre del nuevo programa en Basic que será

creado (líneas 260-310). No debe de haber ningún fichero en el disco que lleve el mismo nombre. Si alguno de los ficheros no puede ser abierto, aparece un mensaje de error y el programa se da por terminado.

Observa que los comandos Open de las líneas 250 y 300 abren un fichero de programa (P) para leer y escribir, respectivamente. Esto no se puede realizar con el comando 4.0 DOpen en Basic. El comando estandard Open debe ser utilizado tal y como se presenta.

Seguramente no encontrarás esto en la documentación de Commodore, pero los ficheros de programa pueden ser abiertos y utilizados igual que los ficheros secuenciales. Simplemente hay que tener en cuenta que los dos primeros bytes de un fichero especifican la dirección de carga, indicando dónde se cargará el programa en la memoria.

Una vez abiertos los ficheros necesarios, esta utilidad lee y saca en pantalla la dirección de carga del programa en lenguaje máquina (líneas 330-370). Cuando se abre el fichero para leerlo, los dos primeros bytes que se leen representan la dirección de carga en el formato 6502 (byte bajo/byte alto). Por lo tanto, la dirección se convierte en su valor decimal añadiendo el primer byte (el byte bajo) al segundo byte (el alto) multiplicado por 256.

A continuación, una dirección de carga de 1025 se coloca al principio del programa en Basic que se está creando con PRINT#2 en la línea 390. El valor 1025 es la dirección de carga estandard para todos los programas en Basic en los ordenadores Commodore. Si tienes un VIC-20 o un C-64, no hace falta modificar este valor dado que el cargador Basic vuelve a situar el programa cuando se carga en estos equipos. Por lo tanto, este valor es válido para todos los sistemas Commodore.

Entrando en el Bucle Principal

Una vez que el programa de utilidades sabe la dirección inicial; entra en el bucle principal, que lee un byte del programa en lenguaje máquina (línea 410), obtiene el valor decimal del byte (línea 420) y añade los datos a la línea del programa en Basic que se está construyendo en ese momento en L$ (líneas 430-440). También se incrementa el contador de bytes para la longitud del programa en lenguaje máquina.

Se comprueba la longitud de la línea creada en el Programa en Basic en la línea 450 para ver si se pueden añadir más datos. De ser así, el programa vuelve a la línea 410 para leer el siguiente byte del programa en lenguaje máquina. De otra forma, se llama a la subrutina en la línea 580 para añadir esta línea al programa en Basic que se está salvando en disco. Después de escribir la línea en disco, el programa



Cuando el Programa de utilidades lee el último byte del programa en lenguaje máquina y detecta el final del fichero, cualquier dato restante será añadido al fichero en Basic.

vuelve a la línea 400 en vez de la línea 410 para colocar el "token" de los Data (131) al principio de la línea en Basic.

La subrutina de las líneas 580-610 suma la longitud de la línea en L$, más los cinco bytes adicionales para cada línea en Basic a un puntero en LK para calcular el enlace o dirección inicial de la siguiente línea en Basic. El valor del enlace de dos bytes se escribe en el fichero del programa en Basic seguido del número de línea de dos bytes de LN. A continuación, se presenta la línea de L$, junto con un byte de cero como bandera de fin para indicar el final de la línea en Basic. Se utiliza otra subrutina en las líneas 620-630 para convertir la dirección de enlace y el número de la línea en Basic en el formato de direccionamiento de dos bytes 6502 y los escribe en el fichero.

Sólo un Byte más

Cuando el Programa de utilidades lee el último byte del programa en lenguaje máquina y detecta el final del fichero, cualquier dato restante será añadido al fichero en Basic (líneas 470-480). El valor de la variable de estado ST se salva en SS después de cada operación de lectura en el fichero en lenguaje máquina. Este valor será 64 cuando se llega al final del fichero.

Una vez escritos todos los datos en el fichero en Basic, se crea un bucle For.-...Next en L$, insertando la longitud del programa en lenguaje máquina como el número de cuenta del bucle y la lectura de la dirección de carga será utilizada como la dirección del Poke inicial (líneas 500-540).

A continuación se escribe la línea del programa en el fichero en Basic con un enlace de cero (dos bytes, ambos de cero) para indicar el final del programa creado en Basic.

Se ha incluido un pequeño ejemplo del tipo de programa que crea el programa de utilidades. Observa que los números de línea del programa en Basic empiezan en la 10 y se incrementan de 10 en 10. Si quieres modificar el número para la línea inicial, vuelve a definir el valor de LN en la línea 180 de la forma que desees. Del mismo modo, si cambias el valor de LI en la línea 190 se modificará el incremento entre los números de línea.

El programa crea líneas en Basic limitadas a 78 caracteres o menos cuando salen en pantalla. Esto te permite utilizar el editor de pantalla para modificar las líneas si es necesario. Si quieres un resultado más compacto y no te importa no poder editar las líneas, puedes modificar el valor de prueba en la línea 450 de 65 a 250. El valor mayor hará que el programa de utilidades cree líneas de la máxima longitud que permite el Basic. Las líneas pueden salir en pantalla pero no pueden ser editadas. •



```
10 REM ************************
20 REM
30 REM   CONSTRUCTOR DE DATOS
40 REM
50 REM    POR: ROBERT BAKER
60 REM
70 REM  ADAPTADO POR  D.ROMERO
80 REM
90 REM ************************
100 :
170 :
180 LN=10:REM NUMERO DE LA PRIMERA L
INEA BASIC
190 LI=10:REM INCREMENTO DE LINEAS
200 :
210 PRINT"[CLR]EL[SPC]PROGRAMA[SPC]D
E[SPC]LENGUAJE[SPC]MAQUINA"
220 PRINT"A[SPC]CONVERTIR[SPC]ES[SPC]
-":PRINT
230 INPUT F$
240 OPEN15,8,15
250 OPEN1,8,5,"0:"+LEFT$(F$,16)+",P,
R"
260 INPUT#15,EN,EM$:IFEN<>0THENPRINT
"ERROR[SPC]EN[SPC]DISCO-";EN;EM$:GOT
O650
270 PRINT:PRINT"EL[SPC]PROGRAMA[SPC]
GENERADO[SPC]SE[SPC]LLAMARA-":PRINT
290 INPUTF$
300 OPEN2,8,6,"0:"+LEFT$(F$,16)+",P,
W"
310 INPUT#15,EN,EM$:IFEN<>0THENPRINT
"ERROR[SPC]EN[SPC]DISCO-";EN;EM$:GOT
O650
320 PRINT:PRINT"OK,[SPC]CREANDO[SPC]
EL[SPC]NUEVO[SPC]PROGRAMA...":PRINT
330 GET#1,C$:IFST<>0THEN640
340 AD=0:IFC$<>""THENAD=ASC(C$)
350 GET#1,C$:IFST<>0THEN640
360 C=0:IFC$<>""THENC=ASC(C$)
370 AD=AD+(256*C):PRINT"DIRECCION[SPC]
DE[SPC]COMIENZO[SPC]=";AD:PRINT
380 LK=1025:NB=0
390 PRINT#2,CHR$(1);CHR$(4);
400 L$=CHR$(131):REM TOKEN DE 'DATA'

410 GET#1,C$:SS=ST:IFSS<>0THEN470
420 C=0:IF C$<>""THEN C=ASC(C$)
430 IFLEN(L$)>1THENL$=L$+","
440 L$=L$+MID$(STR$(C),2):NB=NB+1
450 IFLEN(L$)<65THEN410
460 GOSUB580:GOTO400
470 IFSS<>64THEN640
480 IFLEN(L$)>1THENGOSUB580
490 IFNB=0THEN650
500 REM LAS SIGUIENTES LINEAS CREAN
UNA LINEA BASIC
510 REM FOR X=0 TO....:READ C:POKE..
.+X,C:NEXT
520 L$=CHR$(129)+"X"+CHR$(178)+"0"+C
HR$(164)
530 L$=L$+MID$(STR$(NB-1),2)+":"+CHR
$(135)+"C:"
540 L$=L$+CHR$(151)+MID$(STR$(AD),2)
+CHR$(170)+"X,C:"+CHR$(130)
550 GOSUB580:PRINT#2,CHR$(0);CHR$(0)
;
560 PRINT"LONGITUD[SPC]=";NB+1;"BYTE
S":PRINT
570 PRINT"CONVERSION[SPC]TERMINADA":
GOTO650
580 L=LEN(L$):LK=LK+5+L:X=LK:GOSUB62
0
590 X=LN:GOSUB620:LN=LN+LI
600 FORX=1TOL:PRINT#2,MID$(L$,X,1);:
NEXT
610 PRINT#2,CHR$(0);:RETURN
620 X1=INT(X/256):X2=X-(X1*256)
630 PRINT#2,CHR$(X2);CHR$(X1);:RETUR
N
640 PRINT:PRINT"ERROR[SPC]EN[SPC]DIS
CO,[SPC]PROGRAMA[SPC]INTERRUMPIDO"
650 CLOSE1:CLOSE2:CLOSE15
```

Clave para interpretar los listados

Todos los listados que se publican en esta Revista han sido ejecutados en el modelo correspondiente de la gama de ordenadores COMMODORE. Para facilitar la edición de los mismos en la Revista y para mejorar su legibilidad por parte del usuario, se les ha sometido a ciertas modificaciones mediante un programa escrito especialmente para ello. Para los programas destinados a los ordenadores VIC-20 y COMMODORE 64, en los que se usan frecuentemente las posibilidades gráficas que aparecen del teclado, se han sustituido los símbolos gráficos que aparecen normalmente en los listados por una serie de letras entre corchetes [] que indican la secuencia de teclas que se deben pulsar para obtener el carácter deseado. A continuación se da una tabla para aclarar la interpretación de las indicaciones entre corchetes:

[CRSRD]= Tecla cursor hacia abajo (sin SHIFT)
[CRSRU]= Tecla cursor hacia arriba (con SHIFT)
[CRSRR]= Tecla cursor a la derecha (sin SHIFT)
[CRSRL]= Tecla cursor a la izquierda (con SHIFT)
[HOME]= Tecla CLR/HOME (sin SHIFT)
[CLR]= Tecla CLR/HOME (con SHIFT)

Las indicaciones [BLK] a [YEL] corresponden a la pulsación de las teclas de 1 a 8 junto a la tecla CTRL. Lo mismo sucede con [RVSON] y [RVSOF] respecto a la tecla CTRL y las teclas 9 y 10.

El resto de las indicaciones constan de la parte COMM o SHIF seguidas de una letra, número o símbolo —por ejemplo [COMM+] o [SHIFA]—. Esto indica que para obtener el gráfico necesario en el programa deben pulsarse simultáneamente las teclas COMMODORE (la que lleva el logotipo) o una de SHIFT y la tecla indicada por la letra, el número o el símbolo, en el ejemplo anterior: COMMODORE y + o SHIFT y A, respectivamente.

En los signos gráficos además se cuenta el número de veces que aparece. Por ejemplo, [7 CRSRR] equivale a 7 pulsaciones de la tecla cursor a la derecha y [3 SPC] tres pulsaciones de la barra espaciadora. •

algunos de nuestros programas para los ordenadores personales

commodore



25.000pts.



15.000pts.





8.000pts.



15.000pts.



5.000pts.



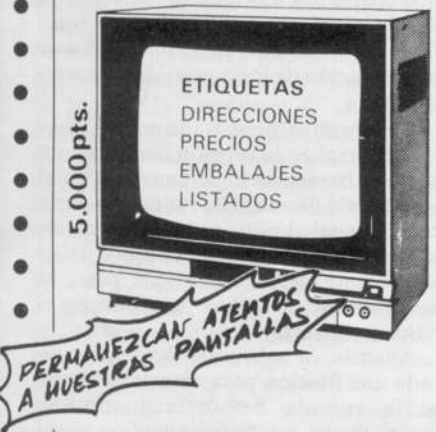
10.000pts.



10.000 pts.



15.000pts.





10.000pts.



10.000pts.



BASIC MICRO-ORDENADORES, S. A.

AVD. CESAR AUGUSTO, 72
Tlfs. 23 56 82 y 22 65 44
50003-ZARAGOZA

Este instrumento importante para la programación del lenguaje de máquina es la clave para examinar las rutinas internas ROM del VIC-20 y el C-64.

El VIC-20 y el C-64 resultan ser muy eficaces para programar la animación, dadas sus maravillosas capacidades para el sonido, el color y gráficas. Todas estas características pueden utilizarse en BASIC, el lenguaje residente del VIC-20 y del C-64. Pero para adquirir una mayor velocidad, es mejor realizar la animación y otras aplicaciones importantes en el lenguaje máquina. Este programa está hecho para el VIC-20 pero también funciona con el C-64 modificando la siguiente línea:

100 DIMB$ (15), A$ (7): POKE 53281, 1: LB$= "[19 SPC]": GOTO 700.

Esto resulta algo problemático para el programador no profesional, dado que los commodore no disponen de un monitor interno para el lenguaje máquina. El hecho de que resulta difícil encontrar una buena documentación sobre la estructura interna del VIC-20 y el C-64 y una descripción pormenorizada de las rutinas ROM complica más el problema.

El programa presentado a continuación es un instrumento de lenguaje máquina que genera vuelcos hexadecimales, vuelcos de ASCII y desensamblajes; además proporciona varias funciones adicionales también importantes para la programación del lenguaje máquina. Con el uso de este programa el usuario puede examinar las rutinas internas del ordenador y así adquirir unos conocimientos de primera mano sobre la estructura de la máquina ¡En cierto modo, el usuario está escribiendo su propia documentación!

VIC - 20

ERRAZQUIN /84

Por Thomas HENRY

Traducido por Valerie SHANKS, y adaptado por Diego ROMERO.

Descubrir el secreto de la estru

Todo el contenido de la memoria se imprime en hexadecimal. Además, los equivalentes ASCII se imprimen en rojo en el lado derecho de la pantalla. Esto resulta conveniente para la búsqueda de mensajes de código.

Sobre el Programa

Estudia el Listado 1 para familiarizarte con la estructura del Instrumento del Lenguaje Máquina. El programa lleva mucha documentación sobre las sentencias REM; en consecuencia, poco hay que decir sobre cualquier sección en especial.

Las líneas 110-140 forman una rutina de conversión de decimal en hexadecimal. Esta rutina se utiliza en muchas partes del programa —principalmente porque la función "PEEK" utilizada en las subrutinas de desensamblaje y vuelco hexadecimal proporciona un resultado decimal. Se introduce la subrutina con D igual al número decimal (entre 0 y 65355) y se convierte en D$ igual al equivalente hexadecimal. Observa el uso de la función CHR$ para la generación de los dígitos correctos (0-9 y A-F).

La subrutina presentada en las líneas 150-210 realiza la función contraria: un número hexadecimal se convierte en el equivalente decimal. La rutina comienza con E$ igual al número hexadecimal de cuatro dígitos y termina con E igual al equivalente decimal. En este caso, la función ASC se utiliza para obtener el número correcto.

Además, la subrutina lleva incorporada una función para detectar errores. Si la entrada hexadecimal contiene algún dígito no hexadecimal, o si su longitud no es exactamente de cuatro dígitos, aparece un mensaje de error y el usuario retorna al bucle principal.

La rutina de una sola línea en la línea 220 simplemente introduce un número hexadecimal. Se genera un campo



hacer la conversión a mano, así que deja que lo haga el ordenador). Lo que hace la rutina realmente es introducir un número y enviarlo a la subrutina en la línea 110.

Como la conversión de hexadecimal a decimal también resulta laboriosa, conviene automatizar las líneas 420-450. De nuevo se recurre a la rutina de utilidad (en la línea 150) para realizar la tarea.

Las líneas 480-540 realizan las rutinas de vuelco hexadecimal y vuelco ASCII. Se representan 20 líneas de vuelco hexadecimal en la pantalla, cada línea teniendo cuatro entradas. Esto significa que las direcciones saltan en grupos de cuatro, lo cual resulta una cifra conveniente para trabajar en hexadecimal.

Todo el contenido de la memoria se imprime en hexadecimal. Además, los equivalentes ASCII se imprimen en rojo en el lado derecho de la pantalla. Esto resulta conveniente para la búsqueda de mensajes de código.

El desensamblaje se incluye en las líneas 550-640. La mayoría de las rutinas utilizadas por el desensamblaje ha sido presentada anteriormente.

Sin embargo, observa que la línea 600 se ocupa de un caso de "código malo", imprimiendo tres interrogaciones consecutivas. La línea 630 llama a la rutina que corresponde a la modalidad de direccionamiento deseada; de esta forma la rutina genera la puntuación adecuada. La variable P sigue la pista de la dirección actual, mientras que C$ es igual al mnemónico (p. ej. LDA, STX, PLA). Más adelante hablaremos de cómo funciona el desensamblaje.

Las líneas 650-690 comprenden la rutina de nivel superior. Las teclas de función programables del ordenador se utilizan para llamar a las distintas características. Por ejemplo, la tecla F1 inicia el desensamblaje y la tecla F5 inicia el vuelco hexadecimal. Las teclas de función generan un código del ASCII de 133 a 137 y las líneas 680 y 690 buscan dicho código.

Estructura y Tablas

La última parte del programa consiste en la estructuración y las tablas. Se utiliza un "array" de variables de "strings" para almacenar el mnemónico necesario. Quizá este tipo de variable resulte un poco más lenta que las demás, pero tiene la ventaja de ocupar relativamente poco espacio. Esto es importante, ya que el VIC-20 estandard sólo dispone de 3.5 K de RAM utilizable. El "array" A$() contiene los 57 mnemónicos mientras que el "array" B$() contiene la información de direccionamiento.

Pondremos un ejemplo para explicar cómo estos "arrays" generan el desensamblaje correcto. Supongamos que un carácter en la memoria resulta ser igual al hexadecimal 25. El dígito más importante (el 2 en este caso) llama al "string" B$(2). Ahora, mirando la línea 590, el dígito menos importante (5) se multiplica por 3 y se incrementa por 1 para dar el número 16. W$ se establece

...tura interior de los Commodore

inverso de cuatro bloques de longitud para indicar al usuario la longitud deseada del número de entrada.

Las líneas 230-360 contienen las subrutinas utilizadas por el desensamblaje. El cometido de estas rutinas es el de generar la "puntuación" adecuada para el campo operando (las comas, signos de dólar, paréntesis, etc.). Dado que el desensamblaje utiliza el mnemónico estandard 6502, cuesta bastante trabajo generar la puntuación más adecuada. Observa que la variable F es de 0,1 ó de 2, según sea la longitud de 0,1 ó 2 caracteres. El "string" A$(0) contiene casi toda la puntuación necesaria, y las secciones deseadas se separan en la línea 250.

Las líneas 270-300 convierten las direcciones relativas en direcciones absolutas al decidir si la bifurcación es regresiva o anticipada, y si está restando o sumando.

La rutina en las líneas 370-410 te proporciona una conversión de decimal a hexadecimal. (Resulta muy laborioso

Se utiliza un "array" de variables de "strings" para almacenar el mnemónico necesario. Quizá este tipo de variable resulte un poco más lenta que las demás, pero tiene la ventaja de ocupar relativamente poco espacio.

igual a MID$(B$[2],16,3), que resulta ser 12C. Este código toma el mnemónico correcto (representado por C$ en la línea 620) a través del segundo mnemónico (AND en este ejemplo) en el "string" A$(1) — lo cual explica el 12 en el "string" 12C.

Se encuentra la modalidad de direccionamiento correcta al tomar ASC ("C"), es decir 67, y restarle 64, dejando 3. Por lo tanto, la subrutina 330 se llama en la línea 630 y se utiliza para generar la puntuación. El código lleva una dirección de página cero en este ejemplo.

Aunque esta rutina parezca un poco larga, en la práctica el programa funciona bastante de prisa. El desensamblaje de 20 líneas tarda seis segundos.

Introducir el Programa

Antes de introducir el Listado 1, ten en cuenta que aunque se vean muchas sentencias REM, no deben de introducirse en el ordenador. Esto sería imposible dada la memoria limitada del VIC-

20. Para que sea más fácil encontrar las sentencias que se tienen que introducir, observa que el programa empieza en la línea 100 y se va aumentando por decenas. Todas las sentencias REM se ubican en otros números. Así que comienza a partir de 100 e introduce todas las líneas que terminen en 0.

Los controles de cursor se utilizan de vez en cuando en las sentencias PRINT y ENTER. Dado que la impresora utilizada para generar este listado no imprime los símbolos de control del cursor (sólo las impresoras Commodore lo hacen), hay que emplear otros medios. Se utilizan las abreviaturas entre paréntesis para indicar los controles del cursor, y los REM en las líneas 31-51 indican cómo deben ser interpretados.

El programa es de fácil introducción, pero hay que tener cuidado que se mecanografíen los espacios exactamente igual que en las distintas sentencias Print.

Después de entrar el programa, es aconsejable salvarlo. Después de comprobar que no hay errores de mecanografía y después de probar el programa, se debe hacer una copia de seguridad.

Utilizar el Programa

Sólo se utilizan las teclas de función [f1] - [f7], para operar el programa.

Primero se carga y se ejecuta el programa. Después de que salga la pantalla con el título, el ordenador espera los mandatos:



• [f1] - Esta tecla inicia el desensamblaje. Después de pulsar la tecla, el ordenador te pedirá una dirección de comienzo. Teclea un número hexadecimal de cuatro dígitos y pulsa la tecla return. Así se inicia el desensamblaje. Si le das una dirección de comienzo errónea, recibirás un mensaje de error.

• [f3] - Esta tecla continúa el proceso de desensamblaje a partir del punto en que tú lo dejaste. Se desensamblan unas 20 líneas adicionales.

• [f5] - Esta tecla inicia el vuelco hexadecimal. Primero se solicitará una dirección de comienzo. Teclea un número hexadecimal de cuatro dígitos y pulsa la tecla return; así se inicia el vuelco hexadecimal. La columna de la izquierda representará la dirección de memoria en azul. Las cuatro columnas siguientes, también en azul, representarán el contenido de la memoria. Las últimas cuatro columnas, en rojo, representarán la interpretación ASCII del contenido de la memoria.

• [f7] - Esta tecla continúa el vuelco hexadecimal a partir del punto en que lo deja el usuario.

• [f2] - Esta tecla permite la conversión de un número hexadecimal en un número decimal. Después de pulsar la tecla [f2] el ordenador solicita un número hexadecimal. Teclea un número de cuatro dígitos y pulsa return. Se representará el equivalente hexadecimal. Si se teclea un número hexadecimal erróneo, se recibirá un mensaje de error.

• [f4] - Esta tecla convierte un número decimal en un número hexadecimal. Después de pulsar la tecla, se solicita un número decimal. Teclea el número y pulsa return. Se representará el equivalente hexadecimal. Ten en cuenta que la entrada tiene que ser entre 0 y 65535 ó, en caso contrario, se representará un mensaje de error.

Para que no tengas que aprender todo de memoria, puedes utilizar una plantilla de cartón para indicar con detalle los usos de las teclas de función. Esta plantilla se demuestra en la Figura 1. Se recorta la plantilla y se pega a un trozo de cartón. Se recorta y se tira la zona del centro. Así la plantilla se acopla alrededor de las teclas de función.

El Instrumento de Lenguaje Máquina resultará útil para entender la estructura interna de ordenador. Mediante el uso del desensamblaje, las rutinas pueden ser interpretadas y se puede descubrir el tipo de estructura necesaria para que éstas puedan ser introducidas. El vuelo ASCII resultará útil tanto para encontrar las palabras clave Basic en la memoria como para los mensajes de error. •

VIC-20

```
10 REM********************************
******
11 REM*
    *
12 REM*  VIC-20 MACHINE LANGUAGE TO
OL   *
13 REM*
    *
14 REM*          THOMAS HENRY
    *
15 REM*     TRANSONIC LABORATORIES
    *
16 REM*        249 NORTON STREET
    *
17 REM*        MANKATO, MN 56001
    *
18 REM*
    *
19 REM*
    *
20 REM*
    *
21 REM*
    *
22 REM********************************
******
23 REM
24 REM
25 REM
30 REM********************************
******
```

```
31 REM*
    *
32 REM*   PAUTAS PARA LA INTERPRETAC
ION  *
33 REM*     DEL LISTADO
    *
34 REM*
    *
35 REM*   SHIFT+CLR         [CLR]
    *
36 REM    HOME CURSOR       [HOME]
    *
37 REM*   CURSOR ARRIBA     [CRSRU]
    *
38 REM*   CURSOR ABAJO      [CRSRD]
    *
39 REM*   CURSOR IZQUIERDA [CRSRL]
    *
40 REM*   CURSOR DERECHA    [CRSRR]
    *
41 REM*   CTRL+RVS ON       [RVSON]
    *
42 REM*   CTRL+RVS OFF      [RVSOFF]
    *
43 REM*   CTRL+7            [BLU]
    *
44 REM*   CTRL+3            [RED]
    *
45 REM*
    *
46 REM*   LOS SIMBOLOS ENTRE CORCHET
ES   *
```

MUSIC 64

PERSONAL
COMPUTER
MUSIC





MUSICA CON COMPUTADOR con el Teclado "MUSIC 64"

El teclado "MUSIC 64" se conecta directamente al COMMODORE 64.

Se puede emplear como sintetizador monofónico o bien como teclado de órgano polifónico. Para ello se necesita lo siguiente:

- Un COMMODORE 64.
- Un vídeo monitor o un receptor de televisión.
- Y una unidad FLOPPY DISK o de cassettes.

ALBAREDA le proporciona a usted:

- Un teclado de 4 octavas de Do a Do.
- Un adaptador para realizar el empalme del teclado *con el computador.*
- El software necesario.

EL SINTETIZADOR MONOFONICO

La relación de los instrumentos de los cuales se dispone, está indicada en el vídeo monitor y son los siguientes:

0	Trompeta	7	Instrumentos de cuerda (cuerdas)
1	Brass	8	Piano
2	Clarinete	9	Organo eléctrico I
3	Campanas	A	Organo eléctrico II
4	Flauta	B	Acordeón
5	Guitarra	C	Random (sintetizador)
6	Wha-Bras (cobreswha)		

Modificaciones paramétricas

F1 para alcanzar el parámetro siguiente.
F3 para regresar al parámetro precedente.
F5 para aumentar el valor del parámetro elegido.
F7 para disminuir el valor del parámetro elegido.

EL TECLADO DEL ORGANO POLIFONO

La relación de los instrumentos de los cuales se dispone, está indicada en el vídeo monitor y son los siguientes:

1	Spinete	4	Flauta
2	Acordeón	5	Banjo
3	Campanas tubulares		

Modificaciones paramétricas

F1	Ataque	F2	Volumen
F3	Declive	F4	Traslado
F5	Sostener	F6	Forma de la onda
F7	Soltar	F8	Ciclo de la operación

Deseo recibir información completa del MUSIC-64 sin ningún compromiso por mi parte:

Nombre ....

Dirección ....

Teléf.: ....Ciudad ....

C.P. .... Provincia ....

Soy distribuidor... □ Soy particular... □

(*) Marcar con un asterisco lo que interese.



INSTRUMENTOS
Y ACCESORIOS MUSICALES

C/. Carmen, 19
TARREGA (Lérida) - Teléfonos (973) 31 04 02 - 31 23 51

```
47 REM*  CORRESPONDEN A TECLAS CON S
HIFT *
48 REM* O CON CBM (ABAJO A LA IZQUIE
RDA).*
49 REM* EL NUMERO QUE APARECE ENTRE
LOS  *
50 REM* CORCHETES ES EL NUMERO DE VE
CES  *
51 REM*    QUE DEBE PULSAR LA TECLA.
   *
52 REM*POR EJEMPLO [7SPC] SON 7 ESPA
CIOS.*
53 REM
54 REM
100 DIMB$(15),A$(7):POKE36879,220:LB
$="[19SPC]":GOTO700
105 REM
106 REM
107 REM***CONVERSION DECIMAL A HEXAD
ECIMAL***
108 REM
109 REM
110 D$="":G=1
120 I=INT(D/16):J=D-16*I:D$=CHR$(J+4
8-7*(J>9))+D$
130 G=G+1:D=I:IFG<BTHEN120
140 RETURN
145 REM
146 REM
147 REM***CONVERSION HEXADECIMAL A D
ECIMAL***
148 REM
149 REM
150 E=0:I=0:IFLEN(E$)<>4THEN210
160 J=ASC(MID$(E$,4-I,1))-48
170 IFJ<0ORJ>22THEN210
180 IFJ>9ANDJ<17THEN210
190 E=E+(16↑I)*(J+7*(J>9)):I=I+1:IFI
<>4THEN160
200 RETURN
210 PRINT"[9CRSRL]ERROR[2SPC]":I=0:G
OTO200
215 REM
216 REM
217 REM*** RUTINA DE ENTRADA DE DIRE
CCION ***
218 REM
219 REM
220 PRINT"[HOM]DIRECCION[SPC]INICIAL
":INPUT"[RVSON][4SPC][RVSOFF][6CRSRL]
";E$:GOSUB150:RETURN
225 REM
226 REM
227 REM*** RUTINAS DE DESENSAMBLADO
PARA LOS DIFERENTES MODOS DE DIRECCI
ONAMIENTO ***
228 REM
229 REM
230 F=0:RETURN
240 F=1:D=PEEK(P+1):B=3:GOSUB110
250 C$=C$+MID$(A$(0),3*A+1,3)+D$+MID
$(A$(0),3*A-5,3)
260 RETURN
270 F=1:D=PEEK(P+1)
280 IFD<128THEND=P+D+2:B=5:GOSUB110:
GOTO310
290 D=128-(127ANDD):D=P-D+2:B=5:GOSU
B110:D=P-D+2
300 D=P-D+2:C$=C$+"[SPC]$"+D$:RETURN

310 C$=C$+"[SPC]$"+D$:RETURN
320 F=2:D=PEEK(P+1)+256*PEEK(P+2):B=
5:GOSUB110:RETURN
330 GOSUB320:C$=C$+"[SPC]$"+D$:RETUR
N
340 GOSUB320:C$=C$+"[SPC]$"+D$+",X":
RETURN
350 GOSUB320:C$=C$+"[SPC]$"+D$+",Y":
RETURN
360 GOSUB320:C$=C$+"[SPC]($"+D$+")":
RETURN
365 REM
366 REM
367 REM*** CONVERSION DECIMAL A HEXA
DECIMAL PARA EL USUARIO ***
368 REM
369 REM
370 PRINT"[HOM]";LB$;"[HOM]DECIMAL";

380 INPUTD$:D=VAL(D$):B=5
390 IFD<1ORD>65535THENPRINT"[7CRSRL]
[CRSRU]ERROR":GOTO410
400 GOSUB110:PRINT"[8CRSRL][CRSRU]=[SPC]
$";D$
410 RETURN
415 REM
416 REM
417 REM*** CONVERSION HEXADECIMAL A
DECIMAL PARA EL USUARIO ***
418 REM
419 REM
420 PRINT"[HOM]";LB$
430 INPUT"[HOM]HEX";E$:PRINT"[CRSRU]
";LB$;"[HOM]":GOSUB150:E$=STR$(E):E=
LEN(E$)-1
440 IFI=0THENRETURN
450 PRINT"[HOM]=[SPC]#";MID$(E$,2,E)
:RETURN
455 REM
456 REM
457 REM*** RUTINA DE VOLCADO HEX Y A
SCII ***
458 REM
459 REM
460 GOSUB220
470 P=E
480 PRINT"[CLR][2CRSRD]";
490 FORZ=1TO20:D=P:B=5:GOSUB110:PRIN
TD$;
500 FORY=1TO4:PRINT"[SPC]";:B=3:D=PE
EK(P):GOSUB110:PRINTD$;:P=P+1:NEXT
510 P=P-4:PRINT"[SPC][RED]";:FORG=1T
O4:D=PEEK(P)
520 IFD<32OR(D>128ANDD<161)THEND=46
530 PRINTCHR$(D);:P=P+1:NEXT
540 POKE212,0:PRINT"[BLU]":NEXT:RETU
RN
545 REM
546 REM
547 REM*** RUTINA DE DESENSAMBLADO *
**
548 REM
549 REM
550 GOSUB220
560 P=E
570 PRINT"[CLR][2CRSRD]";
580 FORZ=1TO20:D=P:B=5:GOSUB110:P$=D
$
590 I=INT(PEEK(P)/16):J=PEEK(P)-I*16
:W$=MID$(B$(I),J*3+1,3)
600 IFW$="???"THENC$=W$:F=0:GOTO640
610 I=VAL(LEFT$(W$,1)):J=VAL(MID$(W$
,2,1))
620 C$=MID$(A$(I),J*3-2,3):A=ASC(MID
$(W$,3,1))-64
630 ONAGOSUB230,240,240,330,270,240,
240,340,350,240,240,360
640 PRINTP$+"[2SPC]"+C$:P=P+1+F:NEXT
:RETURN
645 REM
646 REM
647 REM*** RUTINA DE MAXIMO NIVEL **
*
648 REM
```

```
649 REM
650 GETF$:IFF$=""THEN650
660 PRINT"[HOM]";
670 GETF$:IFF$=""THEN670
680 IFASC(F$)<133ORASC(F$)>138THEN670
690 I=139-ASC(F$):ONIGOSUB370,420,48
0,460,570,550:GOTO650
695 REM
696 REM
697 REM*** PREPARACION DE LAS TABLAS ***
698 REM
699 REM
700 PRINT"[CLR][BLU]":PRINT:PRINT"[SPC]
DESENSAMBLADOR[SPC]VIC-20":PRINT
710 PRINT"[5SPC]POR[SPC]THOMAS[SPC]H
ENRY"
720 A$(1)="ADCANDASLBCCBCSBEQBITBMI"
730 A$(2)="BNEBPLBRKBVCBVSCLCCLDCLI"
740 A$(3)="CLVCMPCPXCPYDECDEXDEYEOR"
750 A$(4)="INCINXINYJMPJSRLDALDXLDY"
760 A$(5)="LSRNOPORAPHAPHPPLAPLPROL"
770 A$(6)="RORRTIRTSSBCSECSEDSEISTA"
780 A$(7)="STXSTYTAXTAYTSXTXATXSTYA"

790 B$(0)="23A53J?????????53C13C???5
5A53B13A??????53D13D???"
800 B$(1)="22E53K?????????53F13F???2
6A53I?????????53H13H???"
810 B$(2)="45D12J??????17C12C58C???5
7A12B58A???17D12D58D???"
820 B$(3)="18E12K?????????12F58F???6
5A12I?????????12H58H???"
830 B$(4)="62A38J?????????38C51C???5
4A38B51A???44D38D51D???"
840 B$(5)="24E38K?????????38F51F???2
8A38I?????????38H51H???"
850 B$(6)="63A11J?????????11C61C???5
6A11B61A??44L11D61D???"
860 B$(7)="25E11K?????????11F61F???6
7A11I?????????11H61H???"
870 B$(8)="??68J???????72C68C71C???3
7A???76A???72D68D71D???"
880 B$(9)="14E68K??????72F68F71G???7
8A68I77A??????68H??????"
890 B$(10)="48B46J47B???48C46C47C???
74A46B73A???48D46D47D???"
900 B$(11)="15E46K??????48F46F47G???
31A46I75A???48H46H47I???"
910 B$(12)="34B32J??????34C32C35C???
43A32B36A???34D32D35D???"
920 B$(13)="21E32K?????????32F35F???
27A32I?????????32H35H???"
930 B$(14)="33B64J??????33C64C41C???
42A64B52A???33D64D41D???"
935 B$(15)="16E64K?????????64F41F???
66A64I?????????64H41H???"
940 A$(0)="[7SPC]#$[SPC]$[SPC],X[SPC]
,Y[2SPC]$[2SPC]$[SPC],X)),Y[SPC]($[SPC]
($"
950 GOTO650
```

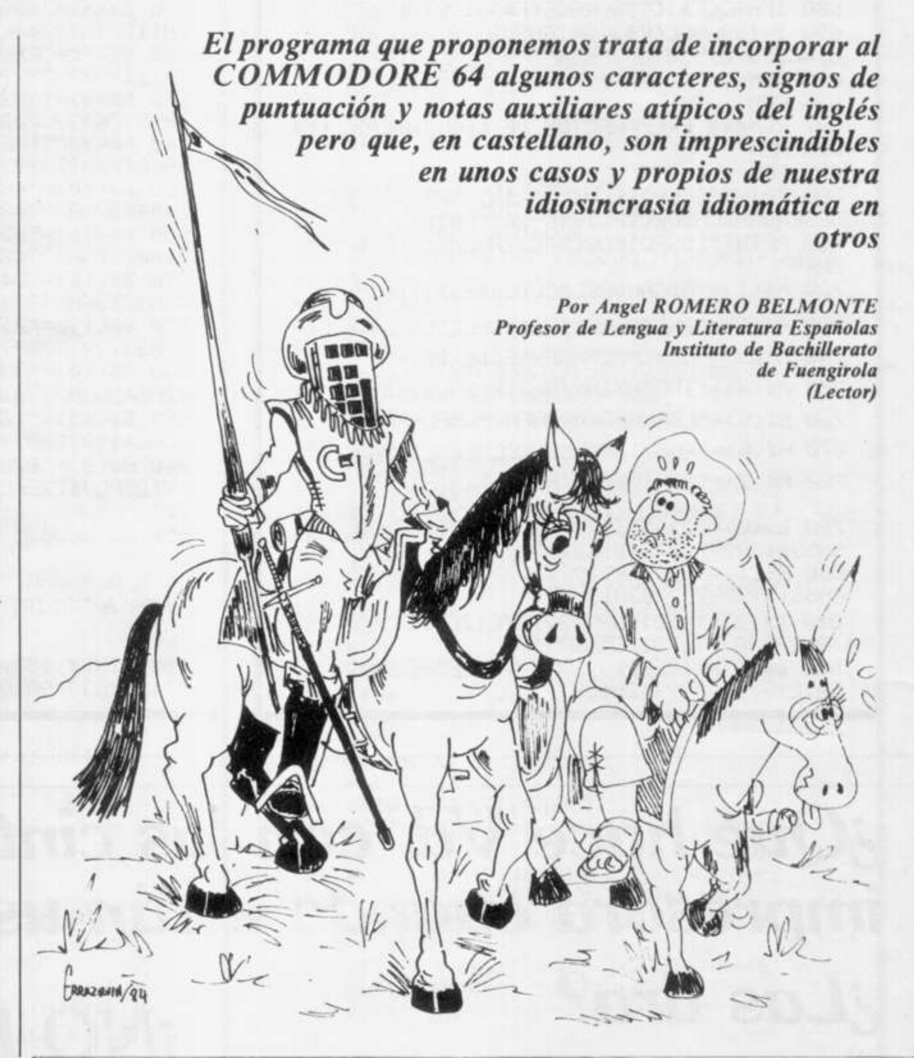
LINEA A MODIFICAR PARA EL C-64

```
100 DIMB$(15),A$(7):POKE53281,1:LB$=
"[19SPC]":GOTO700
```

Castellanizar el Commodore-64

El programa que proponemos trata de incorporar al COMMODORE 64 algunos caracteres, signos de puntuación y notas auxiliares atípicos del inglés pero que, en castellano, son imprescindibles en unos casos y propios de nuestra idiosincrasia idiomática en otros

Por Angel ROMERO BELMONTE
Profesor de Lengua y Literatura Españolas
Instituto de Bachillerato
de Fuengirola
(Lector)



Del **abecedario:** Siguiendo a la Real Academia Española en su opúsculo que recoge las Nuevas Normas de Prosodia y Ortografía (R.A.E., **Ortografía**, Madrid, Imprenta Aguirre, 1974[2]), puede observarse la carencia de la letra "Ñ", en su versión mayúscula y minúscula que, a partir de nuestro programa, se obtienen pulsando las teclas siguientes:



Aunque no es propio del abecedario castellano actual, nos ha parecido interesante añadir el carácter "ç" con el que nuestro ordenador queda "catalanizado" y preparado también para algún uso histórico de nuestro idioma. Se obtiene con C= y C

De los signos de puntuación y notas auxiliares

En castellano, la existencia en algunos casos de una tilde sobre la vocal, además de responder a unas normas ortográficas bien claras y precisas (op. cit., cap. III), que tienden a evitar anfibologías ortográficas (caso de **depósito, deposito, depositó**, por ejemplo), su uso representa al acento de intensidad de la pronunciación y responde perfectamente al "desideratum" de cualquier sistema de escritura: correlato unívoco entre sonido y grafía. Para evitar un programa complejo que supliera la imposibilidad del ordenador (salvo con el uso de "sprites") de sobreimpresionar caracteres (como la máquina de escribir tiene para el caso de " ' "), se ha diseñado un nuevo juego de caracteres vocálicos que, ajustándose estrictamente a las demás formas tipográficas del COMMODORE, incorporan la tilde. Se obtienen pulsando:

C= y la vocal correspondiente.

Admitiendo una costumbre tipográfica que, poco a poco, se va implantando, se excluyen las vocales mayúsculas con acento, pese a que en las **Normas** citadas se insiste expresamente en que "el empleo de mayúsculas no exime de poner tilde..." (párrafo 6, apdo. 15º); sin embargo, su inclusión convertiría el teclado del COMMODORE en un alarde nemotécnico para el programador al tenerse que asignar los caracteres a otras teclas distintas a las vocálicas (por tener éstas ya ocupadas sus tres posibilidades: mayúscula, minúscula, minúscula con tilde). Este es el caso de la forma "ü" (en **desagüe**, por ejemplo) que hemos situado en lugar de @ (pulsación directa) por imposibilidad de incorporarlo en su letra correspondiente. No incluimos la diéresis ("··") en

COMENTARIOS DEL AUTOR

Soy profesor de Lengua y Literatura Españolas en el Instituto de Bachillerato de esta población. Aficionado a la informática, he seguido su revista COMMODORE WORLD desde el comienzo, ello me lleva a enviarles el adjunto artículo para que, si les parece, sea publicado como colaboración desinteresada.
Creo que en nuestro mundillo de "commodorianos" puede resultar interesante el doble objetivo del artículo: cuidar nuestro idioma y "castellanizar" el teclado del ordenador.

EN DEFENSA DEL IDIOMA

No se trata de defender aquí, con un mal entendido purismo academicista, la limpieza de nuestra lengua; todo lo contrario, las lenguas no son productos "incontaminados", sino que, o se hallan genéticamente emparentadas, o se enriquecen por "contagio lingüístico" siguiendo una especie de vaivén histórico, de predominio de unas sobre otras (piénsese cómo, en el terreno concreto de la ciencia, el castellanismo fue el dominante en los siglos XVI-XVII y el galicismo dejó su impronta durante los siglos XVIII-XIX). Estos hechos, inevitables, por otra parte, no tienen nada de censurable. Son formas de enriquecimiento cuando los "préstamos léxicos" no vienen a superponerse a denominaciones ya existentes en el

las demás vocales porque su uso como signo diacrítico indicativo de que un diptongo se ha de considerar como bisilábico, queda reservado a la poesía.

La entonación, expresión de los más diversos estados anímicos o de los más variados sentimientos, se superpone al contenido denotativo de una frase con una flexibilidad y riqueza que otros idiomas no tienen. El castellano posee la particularidad de poder cambiar su esquema dentro de una misma frase; o lo que es lo mismo, en una misma oración puede haber períodos entonativos diversos, de ahí la necesidad real, no capricho academicista, de indicar el punto de la frase en que la entonación pase de una anunciación, por ejemplo, a un tono de sorpresa (admiración) o de pregunta (interrogación); y esta capacidad lo hace necesitar de una representación gráfica de inicio de admiración e interrogación (Vid, como ejemplo la frase: **Entonces, el gran mundo, ¿no le parece a usted mal?**) frente a otros idiomas (por ejemplo el inglés) en los que basta la representación del tonema final por no tener esta flexibilidad. Así pues, los caracteres "¡" y "¿" vienen a sustituir respectivamente a "←" y "↑" del COMMODORE (debe tenerse en cuenta que ahora, el signo de la potenciación "↑" aparecerá en pantalla como "¿" sin que ello implique otras repercusiones.

Los demás signos, o bien ya están incorporados en el ordenador (",", ";"), o bien pueden usarse algunos gráficos sustitutivos. Para la raya ("—") puede usarse [SHIFT] y [*] (ASCII=96); como guión ("-"), el signo de la sustracción (ASCII=45). Otros signos de los incluidos por la R.A.E. como el calderón o la manecilla, han perdido en el uso absoluta vigencia y los dejamos al margen; hacemos una excepción con el párrafo

***H*ay que empezar también, a apreciar, como valor técnico importante, a aquellos ordenadores cuyo teclado venga de origen "españolizado"**

que se incluye en [SHIFT] y [£], y el asterisco ([C=] y [*]), así como las abreviaturas "º", "ª" que pueden obtenerse con [C=] y [@], [SHIFT] y [@], respectivamente.

Mención aparte merece la inclusión de las comillas, que hemos asignado al gráfico [C=] y [*], separándolas así de las "de cadena" que trae incorporadas el C-64 y con lo que se evita el engorro de programarlas con CHR$ (34), como ya saben la mayoría de los usuarios de este ordenador.

La definición de los caracteres

El tema de la definición de caracteres por parte del usuario, ya ha sido tratado en varios sitios, inclusive en esta misma revista (en el núm. 1 se explica el procedimiento y se da una rutina para editar caracteres; también en el núm. 7 hay un programa en el mismo sentido; sin embargo, ambos artículos van dirigidos al VIC 20, sin que se encuentre otra referencia al C-64 que la del núm. 5 con motivo del juego "Los peligros del pozo").

Sin que el programa, por tanto, sea ninguna novedad, nos parece oportuno explicarlo un poco y que pueda servir de base para que alguien añada o cambie alguno de los caracteres propuestos (pensamos, por ejemplo, en la utilidad para estudiantes de teoría de conjuntos, lógica proposicional, de griego, o por qué no, incluso añadir el alfabeto cirílico).

El programa consta de dos partes bien diferenciadas como corresponde a las dos grandes operaciones que hay que realizar para definir nuestros propios caracteres. Un primer grupo de instrucciones para pasar la información que el ordenador tiene sobre los caracteres en la ROM (y que nosotros no podemos cambiar por ser "memoria de sólo lectura") a partir de unas ciertas direcciones de la RAM (en la que ya podemos intervenir cambiando, suprimiendo o añadiendo caracteres); este bloque comprende las líneas 0-7. Una segunda parte del programa (líneas 8-41) en la que se dan las instrucciones pertinentes a los 16 caracteres que nosotros hemos remodelado como queda dicho anteriormente. Expliquemos las cosas con mayor detalle:

Línea 0 → Borra la pantalla y sitúa el letrero de "espera" en el centro de la misma en tanto se ejecuta el programa (unos 35 segundos, aproximadamente).

Línea 1 → Reserva memoria RAM para los caracteres impidiendo que un programa en BASIC pueda invadir esta zona de memoria.

Línea 2 → Inhibe el teclado para evitar inadvertidas interrupciones durante la ejecución del programa que malograrían el mismo.

Línea 3 → Desconecta los registros de E/S y conecta la ROM en su lugar (el VICII, como todos los microprocesadores de su clase, sólo puede procesar simultánea 16K, esto, es, "una página").

idioma originario, sino a suplir algún vacío designativo (siempre muy concreto para el caso de lenguas de gran tradición cultural) motivado por las innovaciones de la ciencia y de la técnica, y sin que esta interferencia produzca reajustes graves al sistema lingüístico propio, entendiendo por esto los que afectan a los subsistemas gramatical y fonológico.

A nadie se le oculta que estamos en "el siglo de la lengua inglesa", especialmente desde su segunda mitad, y que todos los idiomas, de una u otra manera, sufren esta influencia, no siendo el castellano una excepción (al fin de cuentas, tampoco el propio inglés se ha librado de ésto: no se olvide que a lo largo de su historia ha incorporado a su vocabulario un 50% de palabras no genuinas, especialmente del latín).

Desterrados ciertos tópicos como el de que los idiomas son, o tienen, una capacidad especial para expresar unos ciertos campos conceptuales y poca para otros (y ello explicaría el "éxito" de algunas lenguas en determinadas épocas, como queda señalado), hay que buscar la razón de ser de este "colonialismo lingüístico" en otras causas político-económicas, no siempre confesables pero que podrían explicarse con uno de los considerandos del Convenio Multilateral sobre Asociación de Academias de la Lengua Española (presentado en 1960 por la correspondiente colombiana): "...tratándose de los pueblos hispanos, la unidad de lenguaje es uno de los factores que más contribuyen a hacerlos respetables y fuertes en el conjunto de las naciones..."

Y esta búsqueda de unidad es una obligación también del ámbito de la informática "de consumo", muy importante por cuanto es, a la vez, un mundo muy nuevo sin la tradición que en otros terrenos técnicos ha servido para decantar las formas lingüísticas útiles para el castellano, de los neologismos extravagantes e innecesarios introducidos por una mal entendida moda; y muy joven por la edad de sus adeptos, sobre todo por la divulgación a través de su incorporación como asignatura en Academias, Colegios e Institutos, y son ellos los más sensibles a aceptar ciertos cambios influidos por el prestigio de la "letra impresa" de revistas y libros, a los que se consideran como modelos de norma lingüística.

Hemos de empezar a ser críticos receptores de una bibliografía que adopta préstamos de manera desigual: unas veces por aceptación directa de términos ("arrays", "buffer", "stack", etc.). que no sabemos por qué no se traducen; otras, por malas traducciones (como la voz **comando**, del inglés "command", que en castellano tiene otro sentido y se cruza etimológicamente con **mando**); en ocasiones, por creación de neologismos, a partir del inglés, con una indisculpable igno-

Línea 4 —→ Copia en RAM la información sobre los caracteres de origen que posee la ROM y pone el teclado para comenzar en minúsculas (como la posición inicial de una máquina de escribir).

Línea 5 —→ Vuelve a conectar los registros de E/S.

Línea 6 —→ Vuelve a permitir el uso del teclado.

Línea 7 —→ Coloca el "puntero de caracteres" en la nueva dirección de la RAM a la que ha de dirigirse el microprocesador para encontrar la nueva información.

Si el programa se detuviese aquí, tendríamos un juego de caracteres y de gráficos sin ninguna diferencia con el original; pero vamos a crear nuestros propios caracteres. Tomemos el carácter "á" (definido en las líneas 12-13). Hemos de averiguar dos cosas: la dirección de pantalla a la que queremos asignar el carácter, y los datos, en decimal, que definen dicho carácter. Veamos lo primero.

Como se sabe, el C-64 asigna a las teclas 2 juegos de caracteres y 3 posibilidades en cada juego. En uno:

Pulsación directa —→ mayúsculas

[SHIFT] y [Letra] —→ gráfico de la derecha.

[C=] y [Letra] gráfico de la izquierda.

En el otro:

Pulsación directa —→ minúscula

[SHIFT] y [Letra] —→ mayúscula

[C=] y [Letra] —→ gráfico de la izquierda.

Esta última solución es la que hemos adoptado para nuestros propósitos. Tomemos la tecla "A". En pulsación directa nos da "a"; con [SHIFT], tenemos "A" y ahora queremos poner "á"

Caracteres añadidos al Commodore 64

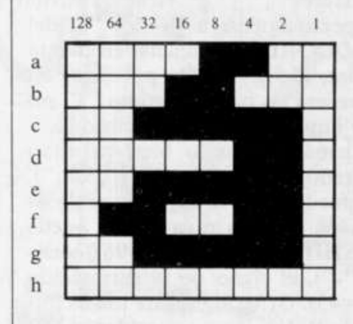
Carácter	Forma de obtenerlo	Código de pantalla	Posición de inicio en RAM
Ñ	[C=] y [M]	103	13112
ñ	[C=] y [N]	106	13136
á	[C=] y [A]	112	13184
é	[C=] y [E]	113	13192
í	[C=] y [I]	98	13072
ó	[C=] y [O]	121	13256
ú	[C=] y [U]	120	13248
ü	[@]	0	12288
ç	[C=] y [C]	124	13280
¡	[←]	31	12536
¿	[↑]	30	12528
§	[SHIFT] y [£]	105	13128
*	[C=] y [*]	95	13048
º	[C=] y [@]	100	13088
ª	[SHIFT] y [@]	122	13264
"	[C=] y [£]	104	13120

en lugar del gráfico de la izquierda. Si localizamos dicho gráfico en el **Manual del Usuario** (pp. 132-134) veremos que le corresponde la cifra 112 como "código de pantalla". Para calcular la dirección de memoria a partir de la cual se va a recoger la información sobre este carácter, aplicamos la siguiente fórmula:

P=12288 + (C*8)

en donde P= nº de posición de inicio en RAM (dato buscado) y C= código de pantalla (112, en nuestro ejemplo). Obtendremos como dirección de memoria la 13184 que es nuestro primer dato.

Obtengamos el segundo dato, es decir, las cifras que recojan la información sobre la forma del carácter deseado. Sobre un cuadrado dividido en 64 partes, dibujamos nuestro carácter y obtenemos el núm. que corresponde a cada línea. Así, por ejemplo, de la línea (a) hemos pintado los cuadrados con valores 8 y 4; de la línea (b), los valores 16 y 8, etc.,



a) 8+4=12
b) 16+8=24
c) 32+16+8+4=60
d) 4+2=6
e) 32+16+8+4+2=62
f) 64+32+8+4=102
g) 32+16+8+4+2=62
h) ningún cuadro=0

Por tanto los datos para definir a "a" son: 12, 24, 60, 6, 62, 102, 62, 0

Ahora se trata de aplicar la siguiente subrutina para situar estos datos en sus lugares de memoria correspondientes:

—FORB=P (posición de inicio) TOP+8 (última posición del carácter):

READC (datos decimales del carácter): POKEB, C: NEXT.

—DATA (8 núm. decimales del carácter)

Para nuestro ejemplo, tendríamos:

```
12 FORB=13184 TO 13191: READC: POKEB, C: NEXT
13 DATA 12, 24, 60, 6, 62, 102, 62, 0
```

Definidos todos los caracteres (16 en nuestro programa) y ordenadas las líneas en sucesión, tendríamos el listado nº 1.

rancia de los mecanismos de creación castellanos (sirva de ejemplo el caso de la palabra **indexado**, del inglés "index", olvidando que el término lo tomó el inglés del nominativo latino, pero que la base patrimonial del castellano procede del acusativo, con lo que el término, sin necesidad de recurrir a otras lenguas, ha de ser **indiciado**, del lat. **índice (m)**, como lo es el cultismo **índice**. Los ejemplos podrían multiplicarse abundantemente y no es este el propósito.

Hay que empezar también, a apreciar, como valor técnico importante, a aquellos ordenadores cuyo teclado venga de origen "españolizado" (que sepamos, algunos microprocesadores ya lo traen incorporado), presionando así a los fabricantes e importadores a plantearse seriamente la deficiencia.

Entre tanto ocupamos el lugar que nos corresponde en el mundo de la ciencia y de la técnica actual, "...la defensiva pura se estima como un medio provisional para recobrar la iniciativa que conduzca a la victoria..." (Vid, S. GILI GAYA, "El lenguaje de la ciencia y de la técnica", en **Presente y futuro de la lengua española**, Madrid, Instituto de Cultura Hispánica, II, 1964).

Para los que ya poseemos nuestro COMMODORE 64, proponemos las siguientes páginas que, en cierto modo, pueden paliar este problema.

Por Angel ROMERO BELMONTE, Profesor de Lengua y Literatura Españolas, Instituto de Bachillerato de Fuengirola (Lector).

```
0 PRINT"[CLR]":PRINT"[10CRSRD][11CRSRR]
ESPERE[SPC]UN[SPC]MOMENTO"
1 POKE52,48:POKE56,48:CLR
2 POKE56334,PEEK(56334)AND254
3 POKE1,PEEK(1)AND251
4 FORA=0TO2047:POKEA+12288,PEEK(A+55
296):NEXT
5 POKE1,PEEK(1)OR4
6 POKE56334,PEEK(56334)OR1
7 POKE53272,(PEEK(53272)AND240)+12
8 FORN=0TO15
9 READA,B
10 FORM=ATOB:READC:POKEM,C:NEXTM
11 NEXTN
12 DATA13112,13119,60,0,102,118,126,
110,102,0
13 DATA13136,13143,60,0,124,102,102,
102,102,0
14 DATA13184,13191,12,24,60,6,62,102
,62,0
15 DATA13192,13199,12,24,60,102,126,
96,60,0
16 DATA13072,13079,12,24,0,56,24,24,
60,0
17 DATA13256,13263,12,24,60,102,102,
102,60,0
18 DATA13248,13255,12,24,102,102,102
,102,62,0
19 DATA12288,12295,102,0,102,102,102
,102,62,0
20 DATA13280,13287,0,0,60,96,96,60,2
4,48
21 DATA12536,12543,24,0,0,24,24,24,2
4,0
```

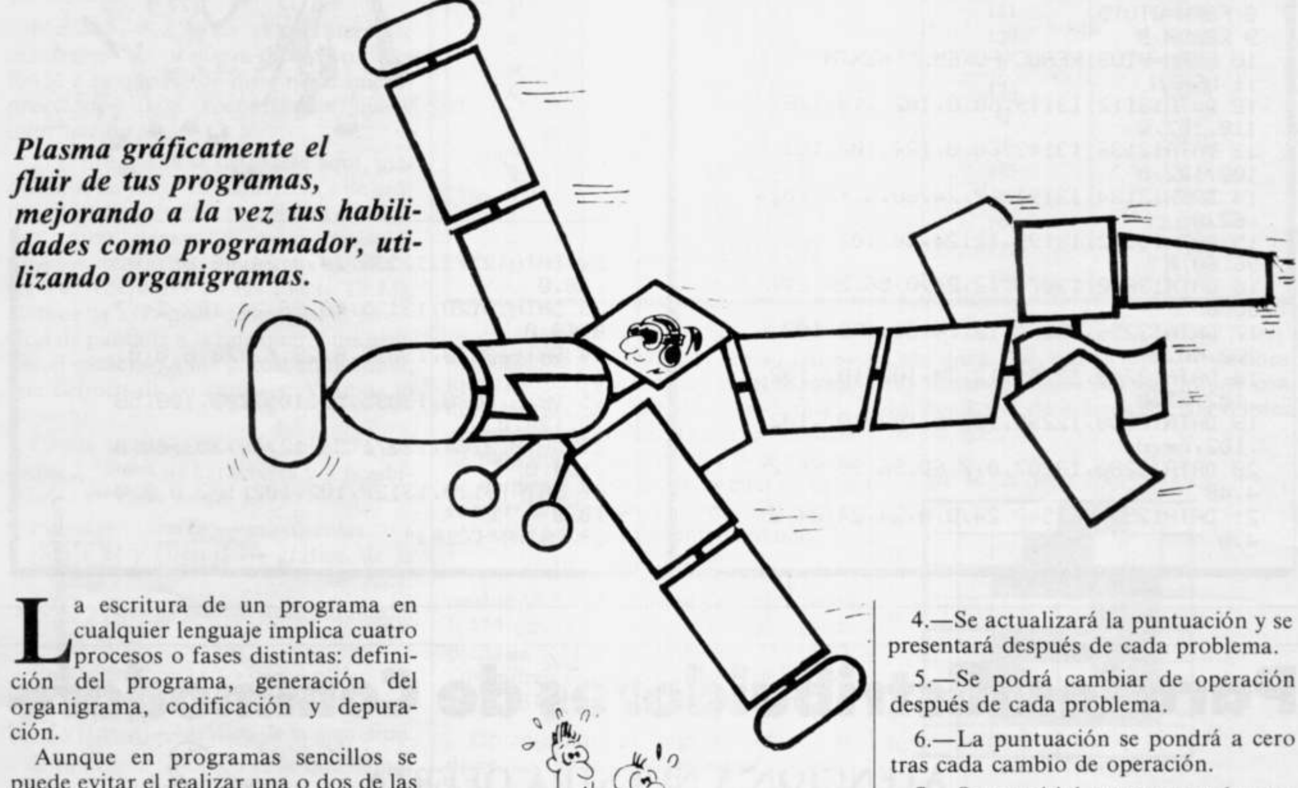


```
22 DATA12528,12535,24,0,24,48,96,102
,60,0
23 DATA13128,13135,60,98,24,102,24,7
0,60,0
24 DATA13048,13055,8,28,62,28,8,0,0,
0
25 DATA13088,13095,56,108,108,108,56
,0,124,0
26 DATA13264,13271,56,12,60,108,60,0
,124,0
27 DATA13120,13127,102,102,102,0,0,0
,0,0
28 PRINT"[CLR]"
```

Por Enoch L. MOSER (RUN EE.UU.)

El arte de realizar organigramas

Plasma gráficamente el fluir de tus programas, mejorando a la vez tus habilidades como programador, utilizando organigramas.



La escritura de un programa en cualquier lenguaje implica cuatro procesos o fases distintas: definición del programa, generación del organigrama, codificación y depuración.

Aunque en programas sencillos se puede evitar el realizar una o dos de las bases sobre el papel, siempre hay que realizarlas todas, aunque sólo sea mentalmente. Cada persona tiene su propia forma de trabajar; algunos son capaces de codificar directamente un programa de gran complejidad.

Sin embargo, la mayoría de los programadores tienen que realizar todas las fases descritas, incluso en los programas más sencillos. Los programas desarrollados ateniéndose a dichos pasos suelen ser, generalmente, de mejor calidad que aquellos que se codificaron directamente.

Definición del Programa

Antes de escribir un programa, lo primero que debes tener claro es lo que quieres hacer. Esto puede parecer obvio pero, como saben los estudiantes de ciencias, el primer paso para resolver un problema es definirlo.

Supón que quieres escribir un programa para que un estudiante realice ejercicios aritméticos. No es suficiente saber que quieres escribir una lección de aritmética. Debes tener en mente un conjunto completo de objetivos y lo mejor para ello es escribirlos.

Tus objetivos deben estar bien detallados, pero no creas que no van a cambiar una vez escritos. Puedes, y debes revisarlos según vaya evolucionando el programa aunque deberías considerarlos inicialmente con cuidado para saber lo que realmente quieres hacer.

Por ejemplo, empecemos con los siguientes objetivos:

1.—Los problemas aritméticos se generarán aleatoriamente.

2.—El estudiante podrá seleccionar cualquiera de las cuatro operaciones aritméticas (suma, resta, multiplicación, y división).

3.—En cada una de las operaciones, el estudiante podrá seleccionar entre cuatro grados de dificultad.

C-64 - VIC-20 SIN AMPLIAR

4.—Se actualizará la puntuación y se presentará después de cada problema.

5.—Se podrá cambiar de operación después de cada problema.

6.—La puntuación se pondrá a cero tras cada cambio de operación.

7.—Se permitirá poner a cero la puntuación sin cambiar de operación.

8.—Cada vez que se resuelva un problema se indicará si la solución es correcta.

9.—Si la respuesta está equivocada se mostrará la solución correcta.

10.—Se generarán informes sobre los progresos del estudiante, basados en la puntuación y el nivel de dificultad.

Una vez definidos los objetivos del programa podemos pasar a la siguiente fase.

Realización del Organigrama

Cualquier programa, no importa en qué lenguaje esté escrito (en código máquina o en un lenguaje de alto nivel como el BASIC), está formado por un conjunto de secuencias de operaciones y un conjunto de puntos de decisión desde los que se seleccionan las secuencias a ejecutar.

Mediante un organigrama (representación gráfica de dichas secuencias y decisiones) se puede visualizar fácilmente el flujo del programa. Si está bien hecho, el programador puede estructurar el programa de forma eficiente.

Mediante un organigrama el autor puede:

—Asegurarse de que se satisfacen todos los objetivos.

—Verificar que se han considerado y previsto todas las posibilidades.

—Identificar el código necesario en varios sitios dentro del programa, y segregarlo si procede en forma de subrutinas evitando, de esta forma, la redundancia de código.

Además, si se utilizan organigramas, se simplifica el resto de los procesos y el programa resultante es más profesional.

Los símbolos utilizados en los diagramas se atienen a ciertas normas, y si es preciso realizarlos con pulcritud, es posible conseguir plantillas especiales sin gran dificultad. Sin embargo, no es totalmente obligatorio tener una plantilla para realizar bien un organigrama, ni tampoco es necesario utilizar todos los símbolos que se han definido. De hecho, muchos programas, si no todos, pueden representarse adecuadamente con tan sólo tres símbolos.

En la Figura 1 se muestran los símbolos utilizados en los organigramas, tomados de una plantilla basada en la norma americana X3.5-1966. Entre ellos, los tres más utilizados son: decisión, proceso, y proceso predefinido y conector.

Los procesos predefinidos, normalmente llamados subrutinas, suelen distinguirse añadiendo una línea vertical u horizontal al símbolo. Esto se utiliza para indicar que dicho símbolo engloba muchos pasos de programas, posiblemente incluyendo operaciones y decisiones. Descubrirás que es muy útil usar estos procesos predefinidos en todos los casos salvo, quizás, en los más sencillos.

En la Figura 2 se muestra una variante del símbolo de decisión, que resulta muy útil cuando existan muchos puntos de salida (por ejemplo, la sentencia ON de BASIC). Esta situación puede representarse, también, por una secuencia de bloques de decisión única; pero esta variante es más compacta y simplifica la visualización del flujo del programa.

Organigramas organizados sin fallos

Cuando se diseña un organigrama, se deben tener en cuenta 4 factores importantes.

En primer lugar, se debe suponer que el organigrama lo va a tener que interpretar otra persona y que tú no vas a estar allí para explicárselo. Tú mismo podrás apreciar los resultados de este supuesto, si uno o dos años después, decides modificar el programa para realizar una nueva aplicación. En ese momento, tú eres la otra persona y no hay nadie que pueda explicarte el organigrama.

En segundo lugar, no incluyas demasiados detalles. La idea es crear una imagen gráfica del programa. Si se incluyen demasiados detalles, se distorsiona la imagen. Los detalles se colocarán en la siguiente fase.

En tercer lugar, cuando hayas concluído el organigrama, revisa los objetivos del programa fijados en la primera fase para asegurarte de que se cumplen. Este puede ser un buen momento para revisar los objetivos pero, asegúrate de que cualquier modificación que realices quede reflejada en el organigrama.

Finalmente, comprueba que se han considerado todas las posibilidades y se han previsto los flujos necesarios. Por ejemplo, ¿qué ocurriría si se realizara una entrada, por teclado, distinta de la esperada? o ¿qué ocurriría si se realizara una división por cero?

Durante la fase de realización del organigrama debes comenzar a elaborar una lista con los nombres de las variables que utilizarás en el programa. Durante esta fase, se suelen detectar las variables necesarias para su utilización como contadores de bucle, indicadores, para registrar las opciones seleccionadas por el operador y para almacenar los datos introducidos mediante el teclado. Los nombres de variables asignados en esta etapa deben utilizarse en el diagrama de flujo e incluirse en la lista de variables. Esto simplificará el proceso de codificación, al reducir el número de veces que deberás repasar la lista.

La creación de la lista de variables continuará en la fase de codificación, ya que, sin duda, necesitarás más variables, cuando llegues al último nivel de detalle.

Esta lista te reportará muchas ventajas posteriormente. Eliminarás el tiempo que normalmente se pierde buscando en el código del programa si un nombre de variable ya ha sido utilizado, o volviendo a codificar partes en las que hayas utilizado nombres de variable previamente asignados.

La figura 3 muestra el diagrama de un programa que cumple los objetivos previamente definidos.

En este programa, como en la mayoría de los programas que forman bucles por naturaleza, la primera ejecución del bucle debe ser diferente del resto. La razón es que deben establecerse las condiciones iniciales. Estas se utilizarán para que el programa comience de forma ordenada.

En este organigrama, el primer bloque refleja el establecimiento de condiciones iniciales. Observa que el flujo del programa tiene en cuenta estas condiciones iniciales y no imprime los indicadores de función o de dificultad en esta primera pasada.

En este momento, se incluyen los nombres de variables que ya se conocen. Se identifican las subrutinas. De hecho, en este momento es posible, e incluso aconsejable, elegir los números de línea de comienzo de las subrutinas. Para el

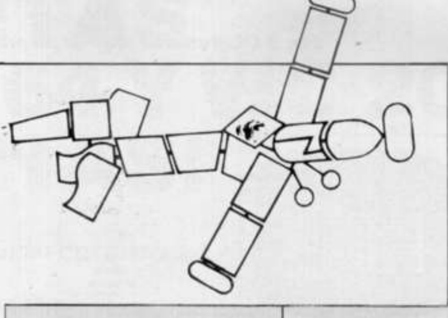




Fig. 1. Símbolos estandard de un diagrama de flujo.

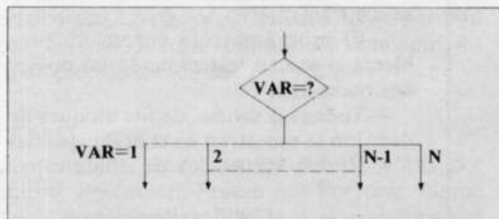


Fig. 2. Símbolo de decisión con salidas múltiples.

VIC-20 y el Commodore 64, no debe importarte utilizar números de línea altos ya que todos ocupan lo mismo en memoria (2 octetos).

La Guía de Referencia del Programador del VIC-20 tiene un error respecto a este punto. El único efecto que sobre la memoria puede tener el utilizar números de línea altos es el acortamiento de

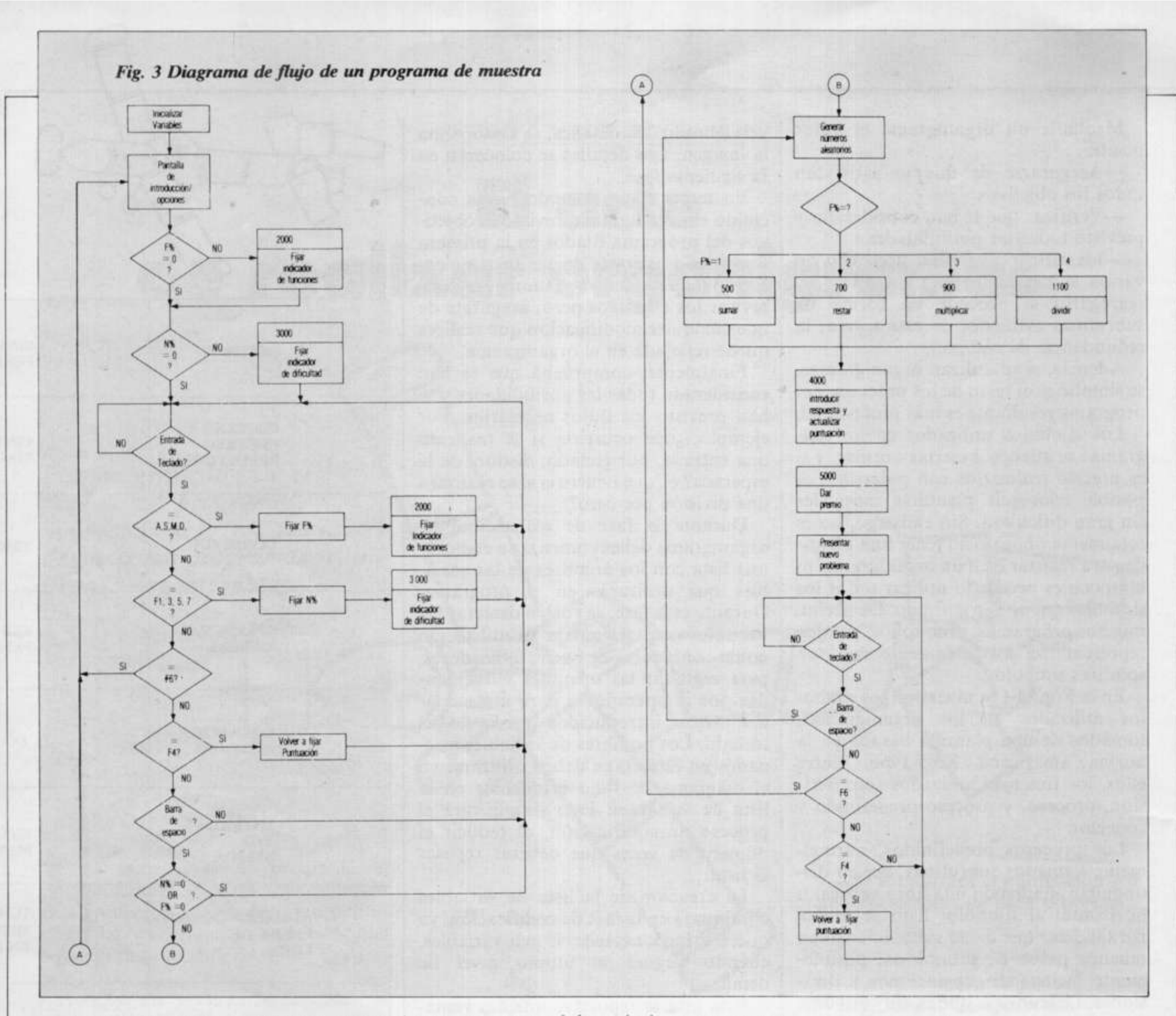


Fig. 3 Diagrama de flujo de un programa de muestra

las líneas a 88 caracteres (80 en el C64) y las referencias en ASCII (un octeto por carácter) a dicha línea. El primer efecto sólo será importante cuando trate de compactar su programa para incluir tantas sentencias como sea posible en una línea.

Otros puntos importantes acerca del organigrama de la Figura 3 son:

—Las entradas incorrectas por teclado se ignoran.

—El programa sólo genera un problema si se han seleccionado las opciones necesarias.

—Todas las salidas de los bloques de decisión se muestran en el organigrama.

—No hay secuencias sin salida.

Codificación

Una vez definidos los objetivos y preparado el organigrama que cubre dichos objetivos, se está en disposición de realizar la codificación propiamente dicha. Este proceso es fácil con el material que ya se ha preparado. Lo único que necesitarás es estar familiarizado con el juego de instrucciones del lenguaje que vas a utilizar, y disponer de una tarjeta de referencia del lenguaje.

Cuando hayas escrito algunos programas no tendrás prácticamente que recurrir a la tarjeta de referencia. Según vaya realizando la codificación, continúa actualizando la lista de variables con aquellos nuevos nombres que vayas asignando.

También es importante documentar internamente el programa, utilizando sentencias de comentario (REM). Esto será especialmente útil si tienes que revisar tu programa dentro de mucho tiempo.

Cuando la memoria es escasa, como ocurre en un VIC sin expansión, y se precisa compactar el programa, se deben eliminar las sentencias REM. En este caso, la alternativa es incluir los comentarios en forma de documentación externa, escribiéndola sobre el listado del programa. Los comentarios del programa que aparecen en este artículo son breves ya que el VIC utilizado no tenía mucha memoria.

Características Especiales

Una vez terminada la codificación, observarás que el programa tiene ciertas características que la distinguen de cualquier otro en el que no se hayan realizado las dos primeras fases. Entre ellas están:

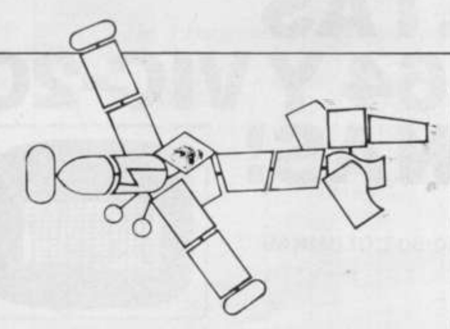
1. Un flujo lógico con números de línea incrementados regularmente.

Cuando el programa se ha codificado directamente, los números de línea no se incrementan regularmente debido a la inserción de objetivos descubiertos sobre la marcha, a la conversión de bloques redundantes en subrutinas o a la adición de sentencias para corregir problemas encontrados en la depuración.

2. Pocos GOTO. Es más fácil añadir nuevas líneas colocándolas fuera del flujo principal del programa y utilizando la sentencia GOTO para entrar y salir, que renumerar todas las sentencias. Aquí se ha utilizado otra técnica para desarrollar el programa.

3. Los segmentos o módulos del programa están definidos con claridad.

Esto es importante cuando es deseable utilizar estos módulos en programas futuros. Si dispones de un amplio conjunto de programas que puedes utilizar en el desarrollo de otros nuevos, obtendrás notables ahorros de tiempo. Los programas que se codifican directamente no son realmente modulares. Esto dificulta tanto el reconocimiento de las líneas que forman los segmentos, como el traslado de éstos a otros programas.



4. Finalmente, la fase restante (depuración) es mucho más simple.

Depuración

La depuración del programa es sencilla si se siguen los pasos que hemos descrito.

En primer lugar, gracias a la labor previa de planificación, no habrá muchos fallos, si bien es cierto que la planificación no puede evitar errores mecanográficos.

En segundo lugar, la estructura lógica, la representación gráfica de dicha estructura y la lista de variables propician el seguimiento del flujo del programa y la detección de cualquier problema. Tendrás que utilizar toda tu habilidad y experiencia en técnicas de depuración para resolver los problemas que puedan aparecer.

Las dos técnicas de depuración más útiles son el comando STOP y la instrucción GOSUB.

Para comprobar si se está ejecutando un segmento determinado o si los valores de las variables son los adecuados, se pueden colocar temporalmente comandos STOP a lo largo del programa. Cuando el programa se detiene a causa de un comando STOP es posible examinar el contenido de las variables, o cambiar sus valores utilizando comandos directos, y luego continuar la ejecución del programa. Esta ejecución no podrá continuar si se cambia una línea, o se añaden o borran líneas después de la parada.

Cuando conviene ejecutar una subrutina GOSUB para comprobar su funcionamiento, puede ser difícil o tedioso conseguir que el programa la llame. Este sería el caso si se intentara depurar la subrutina REWARD en el programa ejemplo. La siguiente inserción temporal en el programa evitaría ésto:

```
1 M=1: F%=1: N%=4: YES=20:NO=0
2 GOSUB 5000
3 STOP
```

Ahora se puede ejecutar este programa temporalmente para comprobar que la subrutina funciona adecuadamente. Las condiciones de la línea 1 pueden cambiarse fácilmente para verificar la subrutina ante distintas hipótesis.

Cuando se hayan realizado todos estos pasos y tu programa esté acabado, le hayas puesto un nombre y lo hayas salvado, debes identificar la documentación producida durante el desarrollo con el nombre del programa y archivarla para su uso posterior.

¿Agobiado?

Un programador sin experiencia puede sentirse agobiado al pensar que debe hacer todo lo que hemos explicado. Sin embargo, es difícil explicar con palabras las ventajas que reporta este método de programación. Cuando se escriben los programas de esta forma, son mejores y se hacen más rápidamente. Sólo la experiencia te convencerá de todo ésto. •

C-64

```
5 REM MATEMATICAS C-64
10 YES=0:NO=0:F%=0:N%=0
20 PRINTCHR$(147)"[2SPC]MATEMATICAS[2SPC]
":PRINT:PRINT:REM OPCIONES
30 PRINT"1.[SPC]ELIJE[SPC]UNA[SPC]OP
ERACION[SPC]PULSANDO:"
40 PRINTTAB(4)"S[SPC]PARA[SPC]SUMAR"

50 PRINTTAB(4)"R[SPC]PARA[SPC]RESTAR
"
60 PRINTTAB(4)"M[SPC]PARA[SPC]MULTIP
LICAR"
70 PRINTTAB(4)"D[SPC]PARA[SPC]DIVIDI
R"
80 PRINT"2.[SPC]ELIJE[SPC]NIVEL[SPC]
DE[SPC]DIFICULTAD[SPC]PULSANDO:"
90 PRINTTAB(4)"F1[SPC]-[SPC]FACIL"
100 PRINTTAB(4)"F3[SPC]-[SPC]NO[SPC]
TAN[SPC]FACIL"
110 PRINTTAB(4)"F5[SPC]-[SPC]DIFICIL
"
120 PRINTTAB(4)"F7[SPC]-[SPC]MUY[SPC]
DIFICIL"
130 PRINT"3.[SPC]PULSA[SPC]F6[SPC](D
ESPUES[2SPC]DE[SPC]CADA[SPC]PROBLEMA
)[SPC]PARA[SPC]CAMBIAR[SPC]DE[SPC]OP
ERACION"
140 PRINT"4.[SPC]PULSA[SPC]F4[SPC]PA
RA[SPC]PONER[SPC]A[SPC]CERO[SPC]EL[8SPC]
MARCADOR"
150 PRINT"5.[SPC]LA[SPC]BARRA[SPC]DE
[SPC]ESPACIOS[SPC]PARA[SPC]COMENZAR"

200 IFF%>0THENGOSUB2000
210 IFN%>0THENGOSUB3000
220 GETA$:IFA$=""THEN220
230 IFA$="S"THENF%=1:YES=0:NO=0:GOSU
B2000:GOTO220
240 IFA$="R"THENF%=2:YES=0:NO=0:GOSU
B2000:GOTO220
250 IFA$="M"THENF%=3:YES=0:NO=0:GOSU
B2000:GOTO220
260 IFA$="D"THENF%=4:YES=0:NO=0:GOSU
B2000:GOTO220
270 IFA$=CHR$(133)THENN%=1:YES=0:NO=
0:GOSUB3000:GOTO220
280 IFA$=CHR$(134)THENN%=2:YES=0:NO=
0:GOSUB3000:GOTO220
290 IFA$=CHR$(135)THENN%=3:YES=0:NO=
0:GOSUB3000:GOTO220
300 IFA$=CHR$(136)THENN%=4:YES=0:NO=
0:GOSUB3000:GOTO220
310 IFA$=CHR$(139)THEN20
320 IFA$=CHR$(138)THENYES=0:NO=0:GOT
O220
330 IFA$=CHR$(32)THEN350
340 GOTO220
350 IFF%=0ORN%=0THEN220
360 FORI=1TO5:RN(I)=INT(RND(X)*10↑N%
):IFF%=4THENRN(I)=RN(I)+1
370 NEXT
380 ONF%GOSUB500,700,900,1100
390 GOSUB4000
400 GOSUB5000
410 PRINT"[HOM][15CRSRD]"
420 PRINT"PULSA[SPC]ESPACIO[SPC]PARA
[SPC]OTRO[SPC]PROBLEMA"
430 GETA$:IFA$=""THEN430
440 IFA$=CHR$(32)THEN360
450 IFA$=CHR$(139)THEN20
460 IFA$=CHR$(138)THENYES=0:NO=0
470 GOTO430
500 RES=0:REM SUMAR
510 PRINTCHR$(147):PRINT"SUMAR":PRIN
T"-----"
520 FORI=1TON%
530 RES=RES+RN(I)
540 PRINTSPC(6)RIGHT$("[6SPC]"+STR$(
RN(I)),N%+2)
550 NEXT
560 RES=RES+RN(N%+1)
570 PRINTSPC(6)RIGHT$("[4SPC]+"+STR$
(RN(N%+1)),N%+2)
580 PRINTSPC(8)RIGHT$("-------",N%)
590 RETURN
700 IFRN(1)>RN(2)THEN720:REM RESTAR
```



¡Hazte socio del Club de Soft de España!
Tendrás multitud de ventajas...

- Recibirás puntual información de las novedades de soft que haya en el mercado. Te enviaremos un catálogo bimensual que incluye más de 200 programas para Commodore 64 y Spectrum.
- Podrás adquirir programas con un 30% de descuento, por ser miembro del CLUB DE SOFT DE ESPAÑA.
- Obtendrás descuentos de hasta un 20% sobre hardware.

Si ya eres socio del Club y deseas acogerte a este regalo, puedes hacerlo contra reembolso de 1.000 ptas., pero no olvides decirnos tu número de socio.

¿COMO PUEDES HACERTE SOCIO DEL CLUB DE SOFT DE ESPAÑA?

- Sólo tendrás que abonar 1.000 Ptas. como cuota UNICA de inscripción.
- Como regalo de bienvenida a nuestro CLUB de SOFT de ESPAÑA podrás elegir entre dos programas:

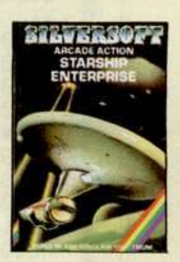
AJEDREZ	STARSHIP ENTERPRISE
SOLO FLIGHT	SLIPPERY SID
para CBM-64	para SPRECTRUM

Te enviaremos el programa elegido completamente GRATIS, junto con la confirmación de tu pertenencia al CLUB de SOFT de ESPAÑA.





CLUB de SOFT de ESPAÑA





DIFUSION POR AMISTAD

Habla a tus amigos de las enormes ventajas que supone pertenecer al CLUB DE SOFT DE ESPAÑA. Si junto con tu inscripción se inscriben dos amigos tuyos por tu recomendación, te **regalamos** un programa de juegos.



CUPON DE INSCRIPCION AL CLUB DE SOFT DE ESPAÑA

Eraso, 12 - Tel. 24610 94 - 28028-MADRID

Nombre ____________________

Dirección ____________________

Ciudad ____________________ Tel. ____________________

Deseo pertenecer al CLUB DE SOFT DE ESPAÑA. Ruego me envíen como regalo de bienvenida al CLUB el programa ____________________

N.º de socio ____________________ Firma

El CLUB DE SOFT DE ESPAÑA te enviará el programa solicitado, junto con un reembolso de 1.000 Ptas. más gastos de envío, como cuota de inscripción al Club.

DIFUSION POR AMISTAD

NOMBRE de un amigo ____________________

Dirección ____________________

Programa de regalo de bienvenida ____________________

Firma

DIFUSION POR AMISTAD

NOMBRE de un amigo ____________________

Dirección ____________________

Programa de regalo de bienvenida ____________________

Firma

```
710 B=RN(1):RN(1)=RN(2):RN(2)=B
720 PRINTCHR$(147):PRINT"RESTAR":PRI
NT"------"
730 RES=RN(1)-RN(2)
740 PRINTSPC(6)RIGHT$("[6SPC]"+STR$(
RN(1)),N%+2)
750 PRINTSPC(6)RIGHT$("[5SPC]-"+STR$
(RN(2)),N%+2)
770 PRINTSPC(8)RIGHT$("-------",N%)
780 RETURN
900 PRINTCHR$(147):PRINT"MULTIPLICAR
":PRINT"-----------":REM MULTIPLICA
R
910 RES=RN(1)*RN(2)
920 PRINTSPC(6)RIGHT$("[6SPC]"+STR$(
RN(1)),N%+2)
930 PRINTSPC(6)RIGHT$("[5SPC]X"+STR$
(RN(2)),N%+2)
950 PRINTSPC(8)RIGHT$("------",N%)
960 RETURN
1100 PRINTCHR$(147):PRINT"DIVIDIR":P
RINT"-------":REM DIVIDIR
1110 IFRN(1)>RN(2)THEN1130
1120 B=RN(1):RN(1)=RN(2):RN(2)=B
1130 RES=INT(RN(1)/RN(2))
1140 RC=RN(1)-RES*RN(2)
1150 PRINTSPC(6+N%);"[COMMM][CRSRD]"
;RIGHT$("[7COMMT][2CRSRU]",N%+3)
1160 PRINTSPC(5)RIGHT$("[6SPC]"+STR$
(RN(1)),N%+1)"[CRSRR]"RIGHT$(STR$(RN
(2)),N%+1)
1170 RETURN
2000 PRINT"[HOM][4CRSRD]";
2010 FORK=1TO4:PRINTTAB(3)"[SPC]":NE
XT
2020 PRINT"[HOM][4CRSRD]";TAB(3+40*(
F%-1))"*"
2030 RETURN
3000 PRINT"[HOM][9CRSRD]";
3010 FORI=1TO4:PRINTTAB(3)"[SPC]":NE
XT
3020 PRINT"[HOM][9CRSRD]";
3030 PRINTTAB(3+40*(N%-1))"*"
3040 RETURN
4000 IFF%<4THENINPUT ANS:RC=0:RI=0:G
OTO4020:REM RESPUESTA
4010 PRINT:INPUT"COCIENTE";ANS:INPUT
"RESTO";RI
4020 IF ANS=RES AND RC=RI THEN YES=Y
ES+1:M=1:GOTO4040
4030 NO=NO+1:M=0
4040 PRINT:PRINT"[7CRSRD]BIEN-";YES,
"MAL-"NO
4050 RETURN
5000 PRINT"[HOM][9CRSRD]"
5010 IFM=1THENPRINT"CORRECTO!!":PRIN
T:GOTO5060
5020 IFF%<4THENPRINT"LO[SPC]SIENTO[SPC]
PERO[SPC]ERA:"RES:PRINT:GOTO5060
5030 PRINT"LO[SPC]SIENTO,[SPC]LA[SPC]
PERO[SPC]LA[SPC]SOLUCION[SPC]ERA:"
5040 PRINTSPC(3)"COCIENTE:";RES
5050 PRINTSPC(3)"RESTO:";RC:PRINT
5060 IFYES=10ANDNO=0ANDN%<4THEN5200
5070 IFYES=20ANDNO=0ANDN%=4THEN5210
5080 IFYES=10ANDNO>2ANDNO<5THEN5220
5090 IFYES=10ANDNO>=5ANDNO<10THEN523
0
5100 IFNO=10ANDYES<NOANDN%>1THEN5240

5110 IFNO=10ANDYES<NOANDN%=1THEN5250

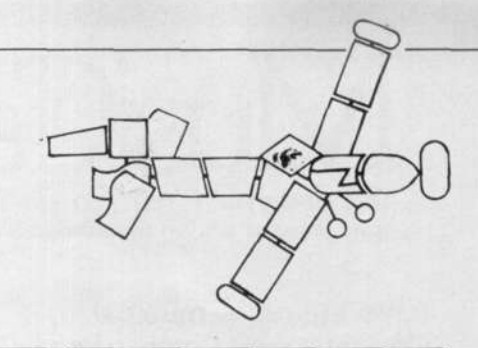
5120 RETURN
5200 PRINT"LO[SPC]ESTAS[SPC]HACIENDO
[SPC]BIEN,POR[SPC]QUE[SPC]NO[7SPC]AU
MENTAS[SPC]LA[SPC]DIFICULTAD[SPC]O[SPC]
"
```

```
5201 PRINT"CAMBIAS[SPC]DE[SPC]OPERAC
ION?[SPC](F6)":RETURN
5210 PRINT"ERES[SPC]DEMASIADO[SPC]BU
ENO[SPC]PARA[SPC]MI.PRUEBA[SPC]CON[SPC]
OTRA[SPC]OPERACION[SPC]O[SPC]DEJA[SPC]
QUE[SPC]";
5211 PRINT"LO[SPC]HAGA[SPC]OTRO":RET
URN
5220 PRINT"LO[SPC]ESTAS[SPC]HACIENDO
[SPC]BIEN,PERO[SPC]TIENES[6SPC]ALGUN
OS[SPC]";
5221 PRINT"FALLOS.[SPC]ESFUERZATE[SPC]
MAS.":RETURN
5230 PRINT"NO[SPC]LO[SPC]HACES[SPC]M
AL,[SPC]PERO[SPC]ESTAS[SPC]COMETIEND
O[2SPC]DEMASIADOS[SPC]ERRORES":RETUR
N
5240 PRINT"NO[SPC]CREO[SPC]QUE[SPC]E
STES[SPC]PREPARADO[SPC]PARA[SPC]ESTE
[3SPC]NIVEL.POR[SPC]FAVOR,ELIJE[SPC]
";
5241 PRINT"UN[SPC]NIVEL[SPC]MAS[SPC]
BAJO.":RETURN
5250 PRINT"POR[SPC]FAVOR,[SPC]PREGUN
TA[SPC]A[SPC]TUS[SPC]PADRES[SPC]O[SPC]
A[SPC]TU[SPC]PROFESOR.":RETURN
```

VIC-20 SIN AMPLIAR

```
5 REM MATEMATICAS VIC-20
10 YES=0:NO=0:F%=0:N%=0
20 PRINTCHR$(147)"[2SPC]MATEMATICAS[2SPC]
":PRINT:PRINT:REM OPCIONES
30 PRINT"1.[SPC]ELIJE[SPC]UNA[SPC]OP
ERACIONPULSANDO:"
40 PRINTTAB(4)"S[SPC]=[SPC]SUMAR"
50 PRINTTAB(4)"R[SPC]=[SPC]RESTAR"
60 PRINTTAB(4)"M[SPC]=[SPC]MULTIPLIC
AR"
70 PRINTTAB(4)"D[SPC]=[SPC]DIVIDIR"
80 PRINT"2.[SPC]ELIJE[SPC]NIVEL[SPC]
DE[5SPC]DIFICULTAD[SPC]PULSANDO:"
90 PRINTTAB(4)"F1[SPC]-[SPC]FACIL"
100 PRINTTAB(4)"F3[SPC]-[SPC]NO[SPC]
TAN[SPC]FACIL"
110 PRINTTAB(4)"F5[SPC]-[SPC]DIFICIL
"
120 PRINTTAB(4)"F7[SPC]-[SPC]MUY[SPC]
DIFICIL"
130 PRINT"3.[SPC]PULSA[SPC]F6[SPC](D
ESPUES[2SPC]DE[SPC]CADA[SPC]PROBLEMA
)PARA[SPC]CAMBIAR[SPC]DE[SPC]OPERACI
ON"
140 PRINT"4.[SPC]PULSA[SPC]F4[SPC]PA
RA[SPC]PONERA[SPC]CERO[SPC]EL[SPC]MA
RCADOR"
150 PRINT"5.[SPC]LA[SPC]BARRA[SPC]DE
[SPC]ESPACIOPARA[SPC]COMENZAR"
200 IFF%>0THENGOSUB2000
210 IFN%>0THENGOSUB3000
220 GETA$:IFA$=""THEN220
230 IFA$="S"THENF%=1:YES=0:NO=0:GOSU
B2000:GOTO220
240 IFA$="R"THENF%=2:YES=0:NO=0:GOSU
B2000:GOTO220
250 IFA$="M"THENF%=3:YES=0:NO=0:GOSU
B2000:GOTO220
260 IFA$="D"THENF%=4:YES=0:NO=0:GOSU
B2000:GOTO220
270 IFA$=CHR$(133)THENN%=1:YES=0:NO=
0:GOSUB3000:GOTO220
280 IFA$=CHR$(134)THENN%=2:YES=0:NO=
0:GOSUB3000:GOTO220
```

```
290 IFA$=CHR$(135)THENN%=3:YES=0:NO=
0:GOSUB3000:GOTO220
300 IFA$=CHR$(136)THENN%=4:YES=0:NO=
0:GOSUB3000:GOTO220
310 IFA$=CHR$(139)THEN20
320 IFA$=CHR$(138)THENYES=0:NO=0:GOT
O220
330 IFA$=CHR$(32)THEN350
340 GOTO220
350 IFF%=0ORN%=0THEN220
360 FORI=1TO5:RN(I)=INT(RND(X)*10↑N%
):IFF%=4THENRN(I)=RN(I)+1
370 NEXT
380 ONF%GOSUB500,700,900,1100
390 GOSUB4000
400 GOSUB5000
410 PRINTCHR$(19)TAB(220)TAB(220)
420 PRINT"PULSA[SPC]ESPACIO[SPC]PARA
[4SPC]OTRO[SPC]PROBLEMA"
430 GETA$:IFA$=""THEN430
440 IFA$=CHR$(32)THEN360
450 IFA$=CHR$(139)THEN20
460 IFA$=CHR$(138)THENYES=0:NO=0
470 GOTO430
500 RES=0:REM SUMAR
510 PRINTCHR$(147):PRINT"SUMAR":PRIN
T"-----"
520 FORI=1TON%
530 RES=RES+RN(I)
540 PRINTSPC(6)RIGHT$("[6SPC]"+STR$(
RN(I)),N%+2)
550 NEXT
560 RES=RES+RN(N%+1)
570 PRINTSPC(6)RIGHT$("[4SPC]+"+STR$
(RN(N%+1)),N%+2)
580 PRINTSPC(8)RIGHT$("-------",N%)
590 RETURN
700 IFRN(1)>RN(2)THEN720:REM RESTAR
710 B=RN(1):RN(1)=RN(2):RN(2)=B
720 PRINTCHR$(147):PRINT"RESTAR":PRI
NT"------"
730 RES=RN(1)-RN(2)
740 PRINTSPC(6)RIGHT$("[6SPC]"+STR$(
RN(1)),N%+2)
750 PRINTSPC(6)RIGHT$("[5SPC]-"+STR$
(RN(2)),N%+2)
770 PRINTSPC(8)RIGHT$("-------",N%)
780 RETURN
900 PRINTCHR$(147):PRINT"MULTIPLICAR
":PRINT"-----------":REM MULTIPLICA
R
910 RES=RN(1)*RN(2)
920 PRINTSPC(6)RIGHT$("[6SPC]"+STR$(
RN(1)),N%+2)
930 PRINTSPC(6)RIGHT$("[5SPC]X"+STR$
(RN(2)),N%+2)
950 PRINTSPC(8)RIGHT$("------",N%)
960 RETURN
1100 PRINTCHR$(147):PRINT"DIVIDIR":P
RINT"-------":REM DIVIDIR
1110 IFRN(1)>RN(2)THEN1130
1120 B=RN(1):RN(1)=RN(2):RN(2)=B
1130 RES=INT(RN(1)/RN(2))
1140 RC=RN(1)-RES*RN(2)
1150 PRINTSPC(6+N%);"[COMMM][CRSRD]"
;RIGHT$("[7COMMT][2CRSRU]",N%+3)
1160 PRINTSPC(5)RIGHT$("[6SPC]"+STR$
(RN(1)),N%+1)"[CRSRR]"RIGHT$(STR$(RN
(2)),N%+1)
1170 RETURN
2000 PRINTCHR$(19)TAB(110);
2010 FORK=1TO4:PRINTTAB(3)"[SPC]":NE
XT
2020 PRINTCHR$(19)TAB(113+22*(F%-1))
"*"
2030 RETURN
3000 PRINTCHR$(19)TAB(242)
```



```
3010 FORI=1TO4:PRINTTAB(3)"[SPC]":NE
XT
3020 PRINTCHR$(19)TAB(242);
3030 PRINTTAB(3+22*(N%-1))"*"
3040 RETURN
4000 IFF%<4THENPRINTSPC(5);:INPUT AN
S:RC=0:RI=0:GOTO4020:REM RESPUESTA
4010 PRINTTAB(22):INPUT"COCIENTE";AN
S:INPUT"RESTO";RI
4020 IF ANS=RES AND RC=RI THEN YES=Y
ES+1:M=1:GOTO4040
4030 NO=NO+1:M=0
4040 PRINTCHR$(19)TAB(220)TAB(66)"BI
EN-";YES,"MAL-"NO
4050 RETURN
5000 PRINTCHR$(19)TAB(220)
5010 IFM=1THENPRINT"CORRECTO!!":PRIN
TTAB(66):GOTO5060
5020 IFF%<4THENPRINT"LO[SPC]SIENTO[SPC]
PERO[SPC]ERA:":PRINTSPC(6)RES:PRINTT
AB(44):GOTO5060
5030 PRINT"LO[SPC]SIENTO,[SPC]LA[SPC]
PERO[SPC]ERA";
5040 PRINTSPC(3)"COCIENTE:";RES
5050 PRINTSPC(3)"RESTO:";RC:PRINTTAB
(22)
5060 IFYES=10ANDNO=0ANDN%<4THEN5200
5070 IFYES=20ANDNO=0ANDN%=4THEN5210
5080 IFYES=10ANDNO>2ANDNO<5THEN5220
5090 IFYES=10ANDNO>=5ANDNO<10THEN523
0
5100 IFNO=10ANDYES<NOANDN%>1THEN5240

5110 IFNO=10ANDYES<NOANDN%=1THEN5250

5120 RETURN
5200 PRINT"LO[SPC]ESTAS[SPC]HACIENDO
[SPC]BIENPOR[SPC]QUE[SPC]NO[SPC]AUME
NTAS[SPC]LA[SPC]DIFICULTAD[SPC]O[SPC]
CAMBIAS"
5201 PRINT"DE[SPC]OPERACION?[2SPC](F
6)":RETURN
5210 PRINT"ERES[SPC]DEMASIADO[SPC]BU
ENO[2SPC]PARA[SPC]MI.PRUEBA[SPC]CON[4SPC]
OTRA[SPC]OPERACION[SPC]O[SPC]DEJA[SPC]
QUE[SPC]";
5211 PRINT"LO[SPC]HAGA[SPC]OTRO":RET
URN
5220 PRINT"LO[SPC]ESTAS[SPC]HACIENDO
[SPC]BIEN,PERO[SPC]TIENES[SPC]ALGUNO
S"
5221 PRINT"FALLOS.[SPC]ESFUERZATE[SPC]
MAS":RETURN
5230 PRINT"NO[SPC]LO[SPC]HACES[SPC]M
AL,[SPC]PERO[SPC]ESTAS[SPC]COMETIEND
O[6SPC]DEMASIADOS[SPC]ERRORES":RETUR
N
5240 PRINT"NO[SPC]CREO[SPC]QUE[SPC]E
STES[5SPC]PREPARADO[SPC]PARA[SPC]EST
E[3SPC]NIVEL.POR[SPC]FAVOR,ELIJE"
5241 PRINT"UN[SPC]NIVEL[SPC]MAS[SPC]
BAJO(F6).":RETURN
5250 PRINT"POR[SPC]FAVOR,[SPC]PREGUN
TA[SPC]A[SPC]TUS[SPC]PADRES[SPC]O[SPC]
A[SPC]TU[5SPC]PROFESOR.":RETURN
```

Le presentamos a SPINNAKER.
Para que también nuestros hijos aprendan jugando con el ordenador.

Muchos padres ya conocen que el ordenador doméstico puede servir para que nuestros hijos jueguen y pasen largos ratos de ocio. Con SPINNAKER los **juegos de ordenador** además sirven **para aprender**.

SPINNAKER es la firma líder mundial en juegos educativos para ordenador. Ahora llega a nuestro país para que también nuestros hijos aprendan jugando con el ordenador.

Nuestros juegos están llenos de emoción, entretenimiento, color, música... Pero además poseen un alto grado de valor educativo, ayudando a los niños a que desarrollen sus habilidades y aprendan de una divertida manera. Aprovechando al máximo las nuevas posibilidades que nos ofrece el mundo de las computadoras.

En su elaboración han intervenido especialistas en la educación, la didáctica, la música, los gráficos y la programación, que han realizado un cuidado producto especialmente pensado para la educación.

Estos son los motivos por los que miles de niños de todo el mundo son verdaderos fans de SPINNAKER. Y sus padres también.

Si Ud. realmente desea que su ordenador familiar sea una ayuda útil para el futuro de sus hijos, recuerde este nombre: SPINNAKER: Una divertida familia de juegos educativos.



DISPONEMOS DE VERSIONES PARA APPLE II, Commodore 64 y Sinclair Spectrum.

Precio especial de introducción 2.400 pts.



Gran Vía Carlos III, 97 K. 08028 BARCELONA



SPINNAKER SOFT COR. CAMBRIDGE, MASS. USA.



Por Jordi SASTRE

Rutinas Basic (3)

Paginación

El tipo más frecuente de listado que se realiza con un ordenador es el típico listado lineal, que se compone de una cabecera y un cuerpo. El cuerpo está formado por un número determinado de líneas (siempre las mismas en cada página), en que cada línea corresponde a un registro. Todas las líneas tienen la misma estructura, aunque sus contenidos son diferentes por pertenecer a diferentes registros.

Cuando se programan estos listados hay dos aspectos que siempre son engorrosos: el primero es calcular, formatear y alinear adecuadamente las columnas del mismo. El segundo es implementar un contador de líneas para paginar en el momento preciso (ni una línea antes ni una después).

La rutina de PAGINACION que presento en este artículo cumple las siguientes funciones: cada vez que se llama (GOSUB 27100) incrementa un contador de líneas (CL). Si este contador no llega a 63 no hace nada más. En cambio, cuando se llega a sobrepasar la línea 63 imprime una serie de interlíneas hasta llegar a las 66 que se supone tiene la hoja de listado, incrementa el contador de páginas (CP) y confecciona la cabecera de listado para continuar listando.

De esta manera, el programador puede olvidarse de la paginación. Sólo debe preocuparse de efectuar el GOSUB después de imprimir cada línea, y la rutina hará todo lo demás.

La rutina de paginación es corta, pero debe manejarse con cuidado, pues maneja bastantes variables:

Y1$, Z1$, TE$ y TF$ son cuatro variables que pueden asignarse una vez durante el programa (por ejemplo al principio del mismo) y después ya no preocuparse de ellas.

**Y1$** debe contener el carácter de expansión de caracteres de la impresora (CHR$(1) en impresoras Commodore). Si la impresora no posee modo expandido, o no interesa hacerlo, puede quedar como cadena nula (Y1$=" ").

Z1$ debe contener 80 guiones que se utilizarán para trazar las líneas horizontales (si se desea puede utilizarse otro carácter).

TE$ puede contener (no pasa nada si es una cadena nula) el nombre de la empresa que utiliza la programación, o el título del programa, o cualquier texto común a toda la programación.

TF$ puede contener la fecha del día (unos 8 caracteres, por ejemplo: DD. MM. AA.).

Estas variables son las comunes a todos los listados que confeccione un programa. Las variables particulares de cada listado, que debe asignarse antes de la primera llamada a la rutina, son las siguientes:

T0$ debe contener el título del listado (máximo 40 caracteres).

T1$ es la primera línea de cabecera.

**T2$** es la segunda línea de cabecera (si sólo hay una T2$=" ").

También antes de la primera llamada a la rutina deben inicializarse los contadores y forzar una primera impresión de la cabecera. Ello se consigue después de asignar T0$, T1$ y T2$, ejecutando lo siguiente:

CL=66: CP=0: GOSUB 27100

EJEMPLO:

Hace dos meses, en el primer artículo de esta serie, cuando hablábamos de la formatación numérica, puse un ejemplo de formatación utilizando un programa que imprimía diez líneas con tres columnas. Reproduzco aquí el mismo programa, pero incluyendo cabecera y rutina de paginación.

Es obvio que cuanto más estrecho sea el listado, más cortos deberán ser los contenidos de las variables TE$ y T0$. Para comprobar eficazmente el buen funcionamiento de la rutina, obligando a imprimir varias hojas, basta con modificar la línea 210 del programa, poniendo el bucle FOR...NEXT a 150, por ejemplo (210 FOR N=1 TO 150), para llenar dos páginas y media. Asimismo también conviene cambiar la línea 130 para que en lugar de elevar a la séptima potencia lo haga sólo a la cuarta (n ↑ 4), puesto que 150 ↑ 7= 1.70859375E+15, lo cual excede de la precisión de la CPU.

Explicación de la rutina

Veamos ahora cómo trabaja la rutina, línea a línea:

27105: Cálculo del ancho del listado, definido por la longitud de T1$

27110: Incrementa contador de líneas. Si no llega a 63 vuelve a la rutina.

27120: Imprime las interlíneas necesarias para llegar al principio de la siguiente página.

27130: Incrementa el contador de páginas.

27135 y 27138: Imprime TE$, la fecha (TF$) y el número de página (CP).

27140 y 27145: Imprime el título del listado centrado, con carácter expandido si Y1$ está asignado o con espacios en blanco intercalados si no lo está.

27150: Imprime la primera línea de cabecera (T1$) si está asignada.

27160: Imprime la segunda línea de cabecera (T2$) si está asignada.

27170: Imprime una línea horizontal como fin de cabecera y retorna.

Observaciones

Esta rutina trabaja de un modo muy peculiar, por lo que su uso sofisticado puede requerir cierta práctica. No obstante, hay algunos puntos que se pueden comentar de antemano.

Los listados no han de ser forzosamente de 80 ó 132 columnas, sino que su ancho es variable. La rutina toma como ancho la longitud de la cadena T1$.

Si el listado sólo tiene una línea de cabecera, la variable T2$ debe dejarse como nula: T2$=" ".

Si no se desea ningún texto de cabecera, la variable T1$ debe asignarse con tantos espacios en blanco como ancho deba ser el listado.

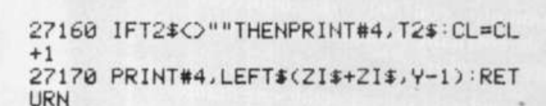
Si no se desea que aparezca el número de página, porque ya se sabe de antemano que el listado sólo tendrá una, puede ponerse CP=-1 en lugar de CP=0 en la inicialización del listado (obsérvese líneas 27130 y 27135). •

Programa "Paginación"

```
10 REM ****************
20 REM *              *
30 REM *  PAGINACION  *
40 REM *              *
50 REM ****************
60 REM
70 :
80 REM   DATOS DE ENTRADA
90 REM   ----------------
100 REM
110 REM T0$ = TITULO DEL LISTADO
120 REM T1$ = LINEA PRIMERA DE CABEC
ERA
130 REM T2$ = LINEA SEGUNDA DE CABEC
ERA
140 REM
150 REN  CONTADORES INTERNOS
160 REM  -------------------
170 REM
180 REM  CL = CONTADOR DE LINEAS
190 REM  CP = CONTADOR DE PAGINAS
200 REM
210 REM  FUNCION DE LA RUTINA
220 REM  --------------------
230 REM
240 REM INCREMENTA EL VALOR DE LAS
         LINEAS.SI SE LLEGA A 63
250 REM COMIENZA UNA NUEVA PAGINA
         IMPRIMIENDO LA CABECERA,

260 REM TITULO DEL LISTADO, NUMERO
         DE PAGINA,...
270 REM
330 REM  OBSERVACIONES
340 REM  -------------
350 REM
360 REM DEBEN ESTAR ASIGNADAS LAS
370 REM VARIABLES Y1$,ZI$,TE$ Y TF$.

380 REM POR EJEMPLO, ANTES DEL PROGR
AMA       DEBE EJECUTARSE:
390 REM   10 ZI$="":FORX=1TO80:ZI$=Z
I$+"-":NEXT
400 REM   20 Y1$=CHR$(1)
410 REM   30 TE$="NOMBRE[SPC]DE[SPC]
EMPRESA[SPC]O[SPC]PROGRAMACION"
420 REM   40 TF$="26.11.84":REM FECH
A
999 :
27100 REM ** PAGINACION ************
***
27105 Y=LEN(T1$)+1
27110 CL=CL+1:IFCL<63THENRETURN
27120 IFCL<67THENFORX=CL+1TO66:PRINT
#4:NEXT
27130 CP=CP+1
27135 PRINT#4:PRINT#4,TE$SPC(Y-20-LE
N(TE$))TF$;:IFCPTHENPRINT#4,"[2SPC]P
AGINA"CP;
27138 PRINT#4:PRINT#4
27140 X1=INT((Y-LEN(T0$)*2)/2):IFY1$
<>""THENPRINT#4,SPC(X1)Y1$T0$:GOTO027
150
27145 PRINT#4,SPC(X1);:FORX=1TOLEN(T
0$):PRINT#4,MID$(T0$,X,1)"[SPC]";:NE
XT:PRINT#4
27150 PRINT#4:CL=6:IFT1$<>""THENPRIN
T#4,T1$:CL=CL+1
```



```
27160 IFT2$<>""THENPRINT#4,T2$:CL=CL
+1
27170 PRINT#4,LEFT$(ZI$+ZI$,Y-1):RET
URN
```

Ejemplo de paginación + formatación

```
1 T0$="EJEMPLO[SPC]DE[SPC]PAGINACION
[SPC]+[SPC]FORMATACION"
2 T1$="[72SPC]"
3 T2$=""
4 Y1$=CHR$(1)
5 TE$="CLUB[SPC]COMMODORE"
6 TF$="26.11.84"
10 Z2$="":FORX=1TO80:Z2$=Z2$+"[SPC]"
:NEXT
15 ZI$="":FORX=1TO80:ZI$=ZI$+"-":NEX
T
20 OPEN4,4:REM LISTADO POR IMPRESORA

30 CL=66:CP=0:GOSUB27100:REM FUERZA
PAGINACION
100 FORN=1TO150
110 B=N:MX=5:GOSUB26299
120 B=N↑7:MX=12:GOSUB26299
130 B=LOG(N):MX=10.5:GOSUB26299
140 GOSUB27100:PRINT#4:NEXT
150 END
26200 REM  * FORMAT.MILES **********
***
26205 MY=INT(MX):MD=INT(MX*10-MY*10)
:IFMD=0THEN26270
26210 X=3:X1=INT(10↑MD+.5):B=SGN(B)*
INT(ABS(B)*X1+.5)/X1:X$="":IFB<0THEN
X$="-"
26215 B2=ABS(B):B1=INT((B2-INT(B2))*
X1+.5):B$=STR$(INT(B2))
26220 B$=RIGHT$(B$,LEN(B$)-1)
26225 IFLEN(X$)+LEN(B$)+INT((LEN(B$)
-1)/3)+MD+1<=MYTHEN26235
26230 B$="":FORX=1TOMY:B$=B$+"$":NEX
T:RETURN
26235 IFLEN(B$)>XTHENB$=LEFT$(B$,LEN
(B$)-X)+"."+RIGHT$(B$,X):X=X+4:GOTO2
6235
26240 IFB1=0THENB1$="'"+LEFT$(ZI$,MD
):GOTO26260
26245 B1$=STR$(B1):B1$="'"+RIGHT$("0
00000000"+RIGHT$(B1$,LEN(B1$)-1),MD)

26255 B1$=LEFT$(B1$+"000000000",MD+1
)
26260 B$=RIGHT$(Z2$+X$+B$,MY-MD-1)+B
1$:RETURN
26270 B=SGN(B)*INT(ABS(B)+.5):B$=STR
$(ABS(B)):B$=RIGHT$(B$,LEN(B$)-1)
26275 IFLEN(B$)>3THENB$=LEFT$(B$,LEN
(B$)-3)+"."+RIGHT$(B$,3)
26280 IFLEN(B$)>7THENB$=LEFT$(B$,LEN
(B$)-7)+"."+RIGHT$(B$,7)
26285 IFB<0THENB$="-"+B$
26290 IFLEN(B$)>MYTHENB$="":FORX=1TO
MY:B$=B$+"$":NEXT:RETURN
26295 B$=RIGHT$(Z2$+B$,MY):RETURN
26299 GOSUB26200:PRINT#4,B$;:RETURN
27100 REM ** PAGINACION ************
***
```

MONITOR TM 01



Monitor de 14 pulgadas de rápido calentamiento con 90º de deflección, exento de corriente estática, tubo preconvergente de color.

Señales de entrada seleccionables mediante interruptor:

—RGB lineal a 1 voltio p-p
—YUV lineal
—RGB TTL

Transformable en pantalla verde. Alta resolución (585 × 436)

P.V.P. 59.500 ptas.

MONITOR TM 80

Monitor de 12 pulgadas fósforo verde P- 31, antirreflexivo.
Alta resolución, 900 líneas en centro de pantalla.

P.V.P. 25.000 Ptas.





Ideales para ser utilizados por una gran gama de micro-ordenadores tanto domésticos como profesionales.

Al accionar el interruptor de encendido los monitores quedan preparados instantáneamente para la recepción de datos.

Estos monitores pueden usarse con:
Commodore, Atari 800, modelos de la BBC, Dragón, Lynx, Oric, Apple II y IIe (con columna RGB de color), Positrón 9000 y Sinclair QL.

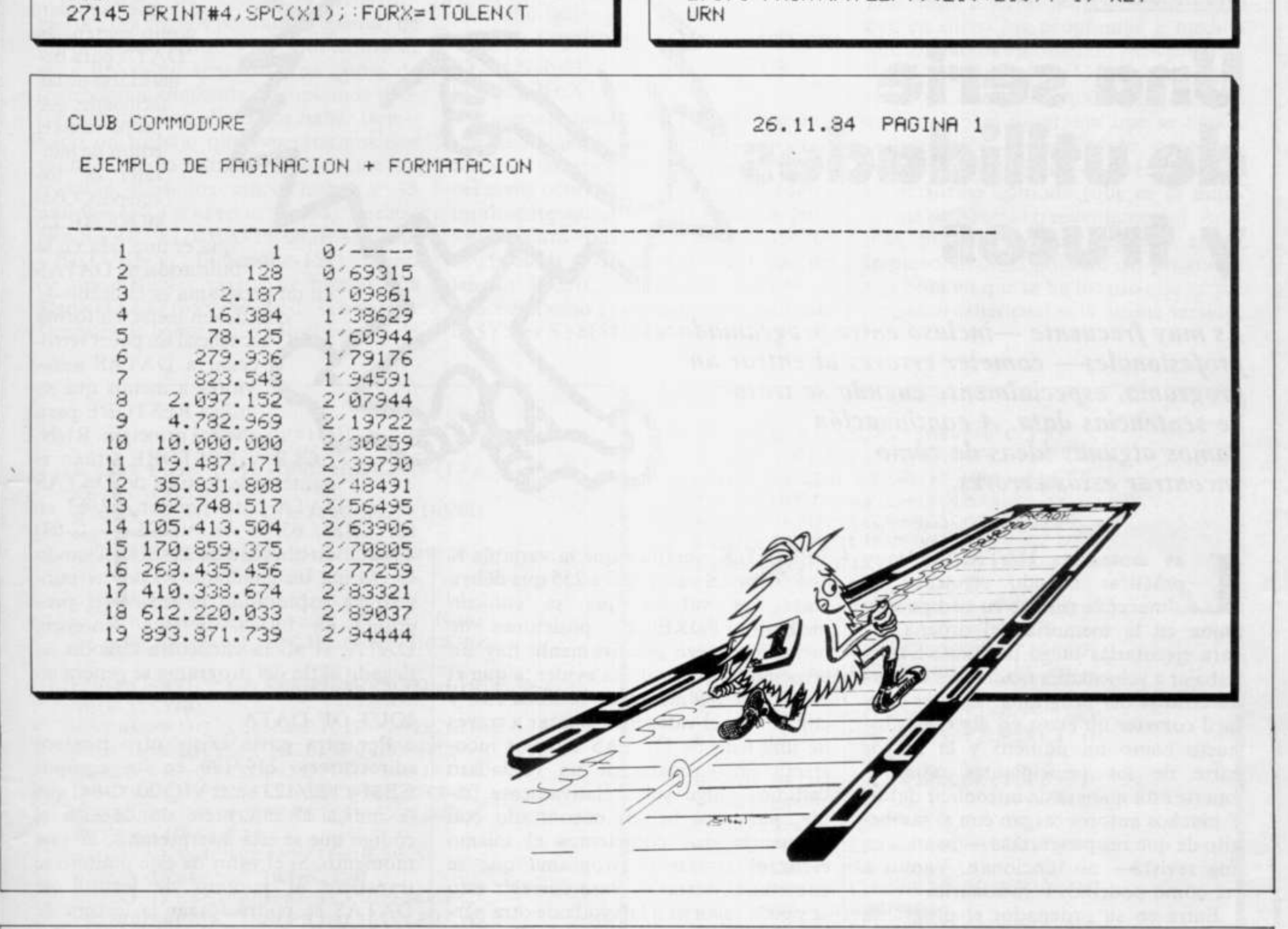
Diseñado y producido por TATUNG. UK (REINO UNIDO)

Importador exclusivo para España. **ALPHA MUNDIAL GROUP, S. A.**

Gran Vía Carlos III, 86-6º
08028 Barcelona

Teléfono 330 96 51
Télex 52220 ARTRA

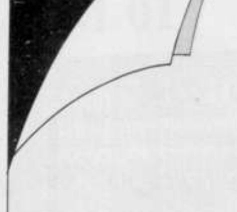
```
27105 Y=LEN(T1$)+1
27110 CL=CL+1:IFCL<63THENRETURN
27120 IFCL<67THENFORX=CL+1TO66:PRINT
#4:NEXT
27130 CP=CP+1
27135 PRINT#4:PRINT#4,TE$SPC(Y-20-LE
N(TE$))TF$;:IFCPTHENPRINT#4,"[2SPC]P
AGINA"CP;
27138 PRINT#4:PRINT#4
27140 X1=INT((Y-LEN(T0$)*2)/2):IFY1$
<>""THENPRINT#4,SPC(X1)Y1$T0$:GOTO27
150
27145 PRINT#4,SPC(X1);:FORX=1TOLEN(T
0$):PRINT#4,MID$(T0$,X,1)"[SPC]";:NE
XT:PRINT#4
27150 PRINT#4:CL=6:IFT1$<>""THENPRIN
T#4,T1$:CL=CL+1
27160 IFT2$<>""THENPRINT#4,T2$:CL=CL
+1
27170 PRINT#4,LEFT$(ZI$+ZI$,Y-1):RET
URN
```



CLUB COMMODORE 26.11.84 PAGINA 1

EJEMPLO DE PAGINACION + FORMATACION

1	1	0'-----
2	128	0'69315
3	2.187	1'09861
4	16.384	1'38629
5	78.125	1'60944
6	279.936	1'79176
7	823.543	1'94591
8	2.097.152	2'07944
9	4.782.969	2'19722
10	10.000.000	2'30259
11	19.487.171	2'39790
12	35.831.808	2'48491
13	62.748.517	2'56495
14	105.413.504	2'63906
15	170.859.375	2'70805
16	268.435.456	2'77259
17	410.338.674	2'83321
18	612.220.033	2'89037
19	893.871.739	2'94444



Por Pere MASATS

Una serie de utilidades y trucos

Es muy frecuente —incluso entre programadores profesionales— cometer errores al entrar un programa, especialmente cuando se trata de sentencias data. A continuación damos algunas ideas de cómo encontrar estos errores.

Las sentencias DATA son muy prácticas cuando se trata de almacenar rutinas en código máquina en la memoria del ordenador, para ejecutarlas luego mediante SYS y trabajar a velocidades razonables en partes críticas del programa. Pero es muy fácil cometer un error en algo tan abstracto como un número y la mayor parte de los principiantes odian a muerte esta manera de introducir datos. Y muchos autores cargan con el sambenito de que sus programas —listados en una revista— no funcionan. Vamos a ver cómo podemos solucionarlo.

Entre en su ordenador el programa siguiente:

```
10 SC=1024: REM PARA EL VIC
   UTILICE 7680 SIN EXPANSION
   O 4096 CON EXPANSION.
20 PRINT" [CLR/HOME]********"
30 FOR J = SC TO SC + 5
40 READ K: POKE J,K
50 NEXTJ
60 REM (MAS LINEAS DE PRO-
   GRAMA)
500 DATA 20,5,19,20
510 DATA 999
```

En este ejemplo se utiliza la conocida técnica de hacer un READ seguido de un POKE en un bucle FOR/NEXT para escribir TEST en la pantalla. El caso es que si hacemos un RUN de este programa nos encontramos con un ?ILLEGAL QUANTITY ERROR IN 40, pero en la línea 40 no hay ningún error. Veamos: si hacemos PRINT J:PRINT K, veremos que la variable K está fuera del valor de 0 a 255 que deben tener los valores que se colocan mediante POKE en posiciones de memoria, luego efectivamente hay un error en la línea 40. Es evidente que el error está realmente en la línea 510 y consiste en el valor 999. Buscar a través de una lista de DATAS el valor incorrecto es —además de un verdadero fastidio— algo sólo relativamente fiable, pues nos hemos encontrado con frecuencia que cometemos el mismo error al revisar el programa que se cometió al entrarlo. Para corregir esto se puede recurrir a la ayuda de otra persona que —con "nuevos ojos"— nos ayude a repasar nuestro programa y el listado. Pero existe una manera muy fácil de encontrar el DATA incorrecto. En las posiciones 63 y 64 (VIC-20 y C-64) se almacena el número de línea del DATA que se ha leído en último lugar, así si después del ?ILLEGAL QUANTITY ERROR hacemos (en modo directo):

```
PRINT PEEK(63)+PEEK(64)*256
```

obtendremos 510 que es la línea donde REALMENTE está el error.

Más sobre datas

De hecho es muy práctico incluir en el programa grandes cantidades de datos que el mismo necesita para su funcionamiento, y muchas veces para mejorar la presentación se recurre a determinadas cadenas que se dan como sentencias DATA para beneficiarse de las facilidades que proporciona el editor de pantalla de los equipos COMMODORE. Lo que es una lata en la utilización de DATAS en un programa es el hecho de que deben leerse en forma secuencial sin poder retroceder a DATAS anteriores a menos que se utilice RESTORE para volver al principio. RUN, CLR y RESTORE sitúan el puntero de lectura de DATAS (DATA read pointer, 62/63 en los CBM, 65/66 en VIC-20 y C-64) apuntando al inicio de BASIC. Cuando se efectúa un comando READ el puntero va explorando el texto del programa en busca de una sentencia DATA, si no la encuentra cuando ha llegado al fin del programa se genera un error
?OUT OF DATA

Por otra parte existe otro puntero (direcciones 119/120 en los equipos CBM y 122/123 en el VIC-20/C-64) que le indica al intérprete dónde está el código que se está interpretando en este momento. Si el valor de este puntero se transfiere al puntero de lectura de DATAS se podrá forzar la lectura de DATAS de forma no secuencial. Por ejemplo:

```
10 DATA PRIMERO, SEGUNDO,
   TERCERO
20 DATA CUARTO
30 READ A$, B$
40 POKE65, PEEK(122): POKE66,
   PEEK(123)
50 READ A, B
60 PRINTA$,B$,,A,B
70 DATA 1,2,3,4
80 END
```

El comando READ de la línea 30 lee en A$ y B$ los DATAS PRIMERO y SEGUNDO dejando el puntero en TERCERO. La línea 40 mueve el puntero a través de las líneas 10, 20 y 30 a un lugar de la misma línea 40, como en esta línea no hay una sentencia DATA el próximo READ se efectuará (línea 50) sobre la línea 70.

En resumen tenemos un sistema de leer los DATAS en el orden que precisemos.

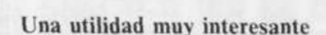
Lectura del canal de error del disco en modo directo

Trabajando con la unidad de disco en un VIC-20 o un COMMODORE 64 suele ser especialmente enojosa la situación siguiente:

Acabamos de depurar el error número... (ya hemos perdido la cuenta) de un programa largo y del que como es muy típico no tenemos una copia de seguridad actualizada, resoplamos profundamente con alivio por haber terminado un trabajo que esperábamos nos tomara media hora y con el que en realidad llevamos cinco horas y 45 minutos (esto si lo recordamos). Tecleamos SAVE "EL RICO PROGRAMA",8 y RETURN y... El bendito 1541 en uno de sus inefables rasgos de humor nos guiña el LED rojo y se queda tan ancho. Después de haber superado el conato de infarto y de buscar infructuosamente un hacha para tomar cumplida venganza (hemos tenido la precaución de esconderla en previsión de situciones como esta) analizamos fríamente la situación; tenemos un (valioso y único) programa en memoria, no sabemos qué error se ha detectado durante la operación de SAVE: puede ser un inocente WRITE PROTECT o un terrible READ ERROR. La solución consiste en leer el canal de error del disco en MODO DIRECTO, para ello teclee lo siguiente:

CLOSE 1: OPEN1,8,15:FORI=OTO30: SYS(43906)#1, A$:?A$;: IFST= OTHENNEXTI.

Y a continuación RETURN (la tecla), inmediatamente y en pantalla nos aparecerá el número, mensaje, pista y sector del error ocurrido, lo que no remediará totalmente nuestros sufrimientos pero los aliviará bastante. Para aquellos usuarios de ordenadores CBM que no tengan BASIC 4.0 en sus equipos el procedimiento es el mismo excepto que el SYS es SYS(51844).



Una utilidad muy interesante

Es una buena precaución —si queremos llegar a viejos programando— grabar en disco los programas a medida que los vamos escribiendo para prevenir fastidiosos cortes de corriente y "gracias" de nuestros retoños, entre otras cosas. Con el programa que se lista a continuación —cargándolo antes de empezar a programar— podremos hacerlo de forma cómoda (que es la única forma de hacerlo frecuentemente). Además podremos hacer listados en la impresora con el nombre del programa y la hora en que se ha listado con lo que será fácil saber cuál es la última versión, etc... •

PROGRAMA: UTILIDAD

```
63000 PG$="UTILIDAD":E$=CHR$(1):
PRINT"[1CLR]"
63010 PRINT"[1HOME][12SPC][1RVSO
N]";PG$
63020 GOSUB63290:PRINT"[1HOME]";
H$;":";M$;":";RIGHT$(TI$,2)
63030 PRINT"[2CRSRD][1RVSON]B[1R
VSOF]ORRAR[1SPC]PRG[1SPC]ANTIGUO
,[1SPC]GRABAR[1SPC]EL[1SPC]NUEVO
."
63040 PRINT"[2CRSRD][1RVSON]L[1R
VSOF]ISTAR
63050 PRINT"[2CRSRD][1RVSON]C[1R
VSOF]AMBIO[1SPC]DE[1SPC]HORA"
63060 PRINT"[2CRSRD][1RVSON]R[1R
VSOF]UN[1SPC]DEL[1SPC]PROGRAMA"
63070 PRINT"[3CRSRD][1RVSON]SELE
CCIONE[1RVSOF]"
63080 GETS$:IFS$=""THEN63020
63090 IFS$="C"THEN63150
63100 IFS$<>"R"ANDS$<>"C"ANDS$<>
"B"ANDS$<>"L"THENPRINT"[1CLR]":E
ND
63110 IFS$="L"THENINPUT"[2CRSRD]
FECHA[1SPC]DE[1SPC]HOY[4SPC]00.0
0.84[10CRSRL]";TD$:GOTO63180
63120 IFS$="R"THENRUN
63130 IFS$="B"THENINPUT"[2CRSRD]
DISCO[1SPC]EN[1SPC]EL[1SPC]DRIVE
[3SPC]0[3CRSRL]";DR$:DR$=DR$+":"
:GOTO63170
63140 PRINT"[1CLR]":END
63150 PRINT"PUESTA[1SPC]EN[1SPC]
HORA:[1SPC]HRMNSG"
63160 INPUT"[16SPC]000000[8CRSRL
]";TI$:GOTO63000
63170 CLOSE1:OPEN1,8,15:PRINT#1,
"S"+DR$+PG$:SAVEDR$+PG$,8:GOTO63
000
63180 OPEN4,4:PRINT#4,E$;CHR$(14
);PG$:PRINT#4
63190 PRINT#4,E$;"VERSION[1SPC]D
EL[1SPC]";TD$;"[2SPC]";:GOSUB632
90:PRINT#4,H$;":";M$+CHR$(15)
63200 INPUT"[1CRSRD]DOBLE[1SPC]A
NCHO[3SPC]N[3CRSRL]";WP$:IFWP$<>
"N"THENPRINT#4,CHR$(15);
63210 INPUT"[1CRSRD]QUIERE[1SPC]
LISTAR[1SPC][1RVSON]T[1RVSOF]ODO
[1SPC]O[1SPC][1RVSON]P[1RVSOF]AR
TE[2SPC]T[3CRSRL]";AP$
63220 IFAP$="T"THENL=0:H=62999:G
OTO63250
63230 INPUT"[1CRSRD]DESDE[1SPC]L
A[1SPC]LINEA[1SPC]NUM.[4SPC]0[3C
RSRL]";L
63240 INPUT"[1CRSRD]A[1SPC]LINEA
[1SPC]NUM.[4SPC]62999[7CRSRL]";H

63250 PRINT"[1CLR]PULSE[1SPC][1R
VSON]RETURN[1RVSOF][1SPC]PARA[1S
PC]LISTAR"
63260 PRINT"[1HOME][5CRSRD]CMD4:
LIST";L;"-";H
63270 PRINT"[1HOME][6CRSRD]PRINT
#4:CLOSE4:GOTO63000"
63280 PRINT"[1HOME][2CRSRD]":END

63290 H$=LEFT$(TI$,2):M$=MID$(TI
$,3,2):H$=STR$(VAL(H$)):RETURN
```





VIC-20 sin expansión

Golf

Joaquín Torres

Recordaréis que en el nº 3 de abril apareció un programa para el C-64 titulado GOLF.

Bueno, pues me he decidido y lo he adaptado para el VIC-20. A ver si podéis mejorarlo, sobre todo en el aspecto musical, porque presentación y juego creo que ya valen.

Yo no he podido poner mejor sonido porque no dispongo de la suficiente memoria como para eso, pues sólo me quedan libres de setecientos a ochocientos bytes.

El programa en sí es similar al del C-64. Sólo cambian los POKE y luego las últimas líneas que son la presentación y la espera de respuestas.

Bueno, ya sólo desearos ánimo y a ver si adaptáis más programas. Y por mi parte seguiré intentándolo.

```
0 REM**********************
10 REM
20 REM   GOLF    VIC - 20
30 REM   ADAPTACION   DE:
40 REM
50 REM   JOAQUIN   TORRES
60 REM
70 REM     02 - 11 - 84
80 REM
90 REM**********************
100 POKE36865,155:POKE36879,8:POKE36
878,15:POKE36869,240:PRINTCHR$(8)
110 PRINT"[CLR][WHT]":GOSUB900
120 V=7680:W=38400
130 FORC=1TO9:Z=176+C
140 PRINT"[CLR]";
150 FORX=0TO21
160 POKEV+(22*18)+X,69
170 POKEW+(22*18)+X,1
180 NEXT
190 PRINT"[8SPC]GOLF[18SPC][4COMMT][9SPC]
"
200 PRINT"EL[SPC]CAMPO[SPC]TIENE[SPC]
20[2SPC]ME-[SPC]TROS."
210 CL=INT(20*RND(0)+1):IFCL<1THEN21
0
220 IFCL>21THEN210
230 POKEV+(22*18)+CL,32
240 POKEV+(22*18)+CL-21,101
250 POKEV+(22*18)+CL-43,101
260 POKEV+(22*18)+CL-65,101
270 POKEV+(22*18)+CL-87,101
280 POKEV+(22*18)+CL-109,Z
290 POKEV+(22*17),81
300 POKEW+(22*17),1
310 PRINT"[HOM][4CRSRD]A[SPC]CUANTOS
[SPC]METROS[SPC]ESTA[SPC]EL[SPC]HOYO
";C;"?"
320 INPUTA:IFA>20THENPRINT"EXAGERADO
![SPC]QUE[SPC]EL[SPC]CAM-PO[SPC]SOLO
[SPC]TIENE[2SPC]20[2SPC]ME-TROS.":GO
TO310
330 IFA<0THENPRINT"NO[SPC]SE[SPC]ADM
ITEN[SPC]TIRADAS[SPC]HACIA[SPC]DETRA
S.":GOTO310
340 FORH=1TOA
350 POKEV+(22*17),32
360 POKEV+(22*17)+H,81
370 POKEW+(22*17)+H,1
380 POKEV+(22*17)+H,32
390 NEXTH
400 POKEV+(22*17)+A,81
410 POKEW+(22*17)+A,1
420 IFA=CLTHEN440
430 GOTO560
440 POKEV+(22*19)+A,81
450 POKEW+(22*19)+A,1
460 POKEV+(22*17)+A,32
470 FORAZ=7680TO7966:POKEAZ,32:NEXTA
Z
480 PRINT"[HOM][2SPC]O.K.[SPC]HOYO"C
"HECHO."
490 REM **ACIERTO**
500 FORL=1TO15
510 FORM=220+L*2TO200STEP-1
520 POKE36876,M
530 NEXTM,L
540 POKE36876,0
550 GOTO720
560 FORAZ=7680TO7966:POKEAZ,32:NEXTA
Z
570 PRINT"[HOM][2CRSRD]EL[SPC]HOYO[SPC]
ESTABA[SPC]A[3SPC]"CL"[2SPC]METROS."
```

Nuevo curso de ordenador personal y programación Basic

SI VAS A SEGUIR VIVIENDO EN ESTE PLANETA NO TE QUEDES DESCONECTADO

¿Cómo dominar a los ordenadores antes de que ellos dominen todos los sectores de la vida?

Si dominas tu impaciencia, ya tienes medio curso hecho



Nuevo curso a distancia Ordenador Personal y Programación Basic

Ahora puedes hacerlo de una forma sencilla y divertida. Con un dedo de la mano. Sin saber una palabra de inglés. Y sin moverte de casa.

Gracias a un curso claro y entretenido que te ayuda a sacarle al ordenador todo lo que tiene dentro.

Por supuesto que no te estamos hablando de un Manual de Instrucciones, y allá tú. Se trata de un programa de aprendizaje muy práctico, diseñado por especialistas, para que nadie se aburra a mitad de camino.

Es, para que te hagas una idea, como un viaje organizado al futuro. ¿Quieres una plaza?

Te ofrecemos 12 manuales de lecciones explicadas paso a paso.

Te ofrecemos, si no tienes ordenador, el COMMODORE 64, el más interesante del mundo, para que practiques desde el primer momento. Te ofrecemos un cuadro de profesores que te orientan continuamente en tu aprendizaje. ¿Qué pones tú? Paciencia. Y un poco de voluntad.

Con esos dos instrumentos te garantizamos que en seis meses serás capaz de operar con cualquier programa que adquieras y algo que te gustará más: podrás crear tus propios programas para uso profesional o personal: gestión, administración, archivo, estadística, juegos, estudios... Verás qué divertido es tener paciencia.

Este es un curso diferente a todos los que conozcas. Es serio, pero sabiendo que no eres una máquina. Es divertido, pero sin olvidar que tienes que sacarle partido.

Consta de los siguientes elementos:

- 12 manuales de lecciones, preparados para aprender enseguida sin perderse en teorías.
- 1 anexo de equivalencias.
- Optativo: Un ordenador personal COMMODORE 64, con su unidad de casettes para practicar desde el primer minuto, porque como se aprende a dominar los ordenadores es trabajando con ellos.
- Amplio número de programas y una orientación continuada de tus profesores.
- No son necesarios conocimientos especiales de ningún tipo.
- Duración = Seis meses aproximadamente dedicando dos horas diarias.
- Sistema de aprendizaje: A distancia.

Cada manual incluye ejercicios amenos para realizar en el ordenador. El profesor contesta a vuelta de correo con las correcciones y los consejos necesarios.

- Precio: El curso completo por algo más del precio del propio ordenador. Y con facilidades de pago.



Centro de Enseñanza a Distancia autorizado en aplicación del Decreto 2641/1980.

Otros de los cursos CCC:

- Electrónica.
- Radio, TV.
- Técnico en Energía Solar.
- Instalador Electricista.
- Técnico en Fontanería.
- Graduado Escolar.
- Guitarra.
- Contabilidad.
- Mecánico de motos.
- Dibujante de comics.
- Inglés (con casette o por video).
- Auxiliar Enfermera.
- Puericultura.
- Estheticiene.
- Peluquería.
- Masaje y Digitopresión.
- Corte y Confección.

Para hablar con los dedos, te echamos una mano

El idioma del futuro, se practica con los dedos. Si quieres aprenderlo, la oportunidad está en tu mano: Envía este cupón.

Deseo información **gratis y sin compromiso** sobre el curso de:

____

Nombre y apellidos ______________________________

Dirección ______________ N.º ______ Piso ______

Población ______________________________

Provincia ______________ Cod. Post. ______

Teléfono ______________ Edad ______

Envía este cupón a:

CCC, alto de Miraconcha - Apdo. 666 - 20080 Tfno.: (943) 467600 - SAN SEBASTIAN o a **CCC**, Apdo. 17.222 - 28080-MADRID

C6-41-83-01-85



```
580 REM **FALLO**
590 FORL=1TO15
600 FORL=230TO200STEP-1:FORM=1TO50:N
EXTM
610 POKE36876,L
620 NEXT
630 POKE36877,220:POKE36876,0
640 FORL=15TO0STEP-1
650 POKE36878,L
660 FORM=1TO300
670 NEXTM,L
680 POKE36877,0
690 POKE36878,15
700 POKE36876,0
710 PRINT"[CRSRD][3SPC]MALA[SPC]SUER
TE,[SPC]OTRA[2SPC]VEZ[SPC]SERA.":GOT
O830
720 IFC<9THENNEXTC
730 PRINT"YA[SPC]NO[SPC]HAY[SPC]MAS[SPC]
HOYOS"
740 FORL=1TO20
750 FORM=254TO240+INT(RND(1)*10)STEP
-1
760 POKE36876,M
770 NEXTM
780 POKE36876,0
790 FORM=0TOINT(RND(1)*100)+120
800 NEXTM,L
810 PRINT"[CRSRD]QUIERES[SPC]OTRA[SPC]
DESDE[SPC]EL[SPC]PRINCIPIO?":GOSUB86
0
820 GOTO130
830 FORM=1TO2000:NEXT:POKE36879,27:P
RINT"[CLR][BLU]":CHR$(9):END
840 GOTO130
850 REM *SUBRUTINA DE PREGUNTA
860 GETB$:IFB$=""THEN860
870 IFB$="S"THENRETURN
880 POKE36879,27:PRINT"[CLR]";CHR$(9
):END
890 REM *SUBRUTINA DE PRESENTACION
900 PRINT"[CLR][2CRSRD][4CRSRR]JOAQU
IN[SPC]TORRES"
910 PRINT"[HOM][5CRSRD][7CRSRR]PRESE
NTA"
920 PRINT"[6CRSRD][RVSON][SHIFT£][RVSOFF]
[2COMM+][RVSON][COMM*][RVSOFF][2SPC]
[RVSON][SHIFT£][RVSOFF][2COMM+][RVSON]
[COMM*][RVSOFF][2SPC][COMM+][5SPC][RVSON]
[SHIFT£][RVSOFF][2COMM+][RVSON][COMM*]
[RVSOFF]";
930 PRINT"[COMM+][2SPC][COMM+][2SPC]
[COMM+][2SPC][COMM+][2SPC][COMM+][5SPC]
[COMM+][3SPC]";
940 PRINT"[COMM+][2SPC][COMM+][2SPC]
[COMM+][2SPC][COMM+][2SPC][COMM+][5SPC]
[COMM+][3SPC]";
950 PRINT"[COMM+][5SPC][COMM+][2SPC]
[COMM+][2SPC][COMM+][5SPC][3COMM+][SPC]
";
960 PRINT"[COMM+][SPC][COMM+][RVSON]
[COMM*][RVSOFF][2SPC][COMM+][2SPC][COMM+]
[2SPC][COMM+][5SPC][COMM+][3SPC]";
970 PRINT"[COMM+][2SPC][COMM+][2SPC]
[COMM+][2SPC][COMM+][2SPC][COMM+][5SPC]
[COMM+][3SPC]";
980 PRINT"[COMM+][2SPC][COMM+][2SPC]
[COMM+][2SPC][COMM+][2SPC][COMM+][5SPC]
[COMM+][3SPC]";
990 PRINT"[COMM*][2COMM+][SHIFT£][2SPC]
[COMM*][2COMM+][SHIFT£][2SPC][COMM*]
[2COMM+][SHIFT£][2SPC][COMM+][3SPC]"
;
1000 REM *SUBE PANTALLA A POSICION N
ORMAL
1010 FORAA=155TO38STEP-1
1020 POKE36865,AA
1030 FORBB=0TO125:NEXTBB,AA
1040 FORBB=0TO9999:NEXT
1050 PRINT"[CLR]";:RETURN
```

C-64

Concierto de Aranjuez

José Luis Muñoz Franco
Monterde, 1-7º B
50010 ZARAGOZA

Hola a todos los lectores de esta magnífica revista. Aquí os mando otro programa al que he titulado EL CONCIERTO DE ARANJUEZ. Se trata de un programa totalmente musical que creo que sin ninguna duda os va a gustar a todos. El funcionamiento del programa es el siguiente:

En pantalla aparece el título y el teclado completo de un piano con 3 octavas y con el número de cada nota de la octava. A continuación y una vez terminado completamente el dibujo, se dan 2 opciones: 1: Manual y 2: Automático. Si pulsáis 1, el ordenador hará sonar una nota del consabido Concierto de Aranjuez cada vez que vosotros

GLOSARIO

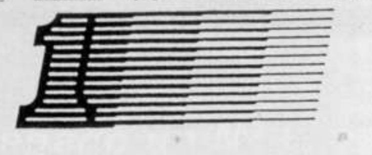
Binary Operator.— Operador binario. Un operador aritmético que relaciona dos término. Los operadores binarios más utilizados son los de: suma (+), resta (—), multiplicación (*) y división (/).

Binary search.— Búsqueda binaria. Es un método de búsqueda dicotómica en el que, en cada paso, se divide el rango a buscar en dos partes iguales y se van realizando aproximaciones consecutivas con los elementos de los extremos del intervalo que contiene al elemento buscado.

BISAM.—Basic indexed sequential access method. Es un método de acceso utilizado en basic para acceder a datos almacenados en unidades que permiten acceso directo, localizándolos rápidamente por medio de un índice que generalmente se guarda en un archivo secuencial junto con los datos. El índice es actualizado cada vez que se hace una modificación que afecte a los campos de búsqueda por clave (ver artículo sobre los ficheros ISAM número 5 de Commodore World).

BIT.—BInary digiT. En notación binario, cualquiera de los caracteres 0 ó 1. Es la unidad más pequeña posible de información. Un bit es suficiente para decir la diferencia entre dos elementos opuestos como sí y no.

Bit pattern.—Configuración de bits. Una configuración de N dígitos binarios que representan una de las 2^N posibles combinaciones con ese número de bits. Por ejemplo con cuatro bits se pueden obtener 16 combinaciones diferentes.



FIRST S.A.

C/ Aribau, 62. BARCELONA- 08011
Tel. (93) 323 03 90
Tlx. 53947 FIRS E (ESPAÑA)

Ordenadores más Personales!



SEGUIMOS BUSCANDO DISTRIBUIDORES

FIRST S.A.

SE COMPLACE EN ANUNCIAR LA DISTRIBUCION DE PRODUCTOS

commodore

DE: microelectrónica y control, s. a. MEE

IMPORTADOR EXCLUSIVO PARA ESPAÑA



ErgoVision
(Base móvil)
29.800 ptas.

NUEVO

22.105 PTAS.

KoalaPad Touch Tablet (disco)



para **COMMODORE 64**

Auténtico tablero gráfico de dibujo para su ordenador. Co nectado puede hacer sus dibujos como lo hace con su lápiz y papel.
Incluye el programa MICRO ILLUSTRATOR de tecnología LISA, permite: Dibujar, Puntos, Lineas, Rayas, Rellenar, Cuadrados, Circulos, Discos, Rellenar, Borrar, Magnificar, Guardará sus gráficos y mucho más. Pásese y compruébelo Vd. mismo.



KoalaPad Touch Tablet

FIRST S.A. importador para españa de Koala

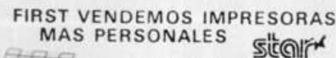


COMMODORE 64
+ 2 SUPER JOYSTICK



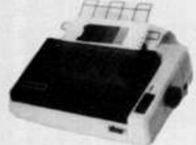
+ .... = PTAS.

METAMORPHIC
25.000 ptas.



FIRST VENDEMOS IMPRESORAS MAS PERSONALES

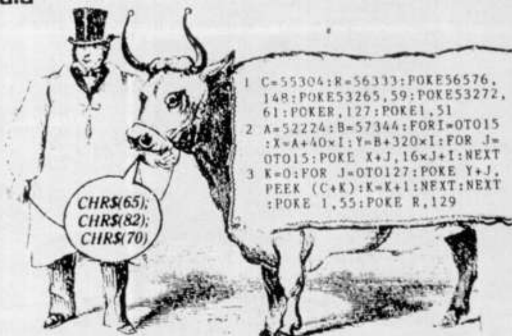
star



MICROANGELO

Es la mas PODEROSA interface paralelo para su impresora Epson Star, etc. existente en el mercado mundial.

GEMINI 10 X... 79.500 Ptas.
GEMINI 15 X... 115.000 Ptas.
INCLUYE LA INTERFACE MICROANGELO Y CONECTORES (OFERTA SOLO PARA ESTE MES)
LLAME PARA SOLICITAR SU PRECIO SI NO PRECISA INTERFACE
LLAME PARA SOLICITAR SU PRECIO PARA EL RESTO DE IMPRESORAS DE LA GAMA STAR. TAMBIEN INCLUYEN LA TARJETA MICROANGELO.




```
1 C=55304:R=56333:POKE56576,
  148:POKE53265,59:POKE53272,
  61:POKER,127:POKE1,51
2 A=52224:B=57344:FORI=OTO15
  :X=A+40×I:Y=B+320×I:FOR J=
  OTO15:POKE X+J,16×J+I:NEXT
3 K=O:FOR J=OTO127:POKE Y+J,
  PEEK (C+K):K=K+1:NEXT:NEXT
  :POKE 1,55:POKE R,129
```





Datalife
10 DISKETTES
+
2 DISKETTES
(LIMPIA CABEZAL)





LAS TARJETAS DE FIRST

- × TARJETA VIC 20 -40/80 COL. LLAME
- × TARJETA VIC 20 -MEMO.64 K. LLAME
- × SLOTS EXPANSION V20 Y C64 LLAME
- × TARJETA COMMO.64 80 COL. LLAME

Más de 200 programas para su VIC 20 y COMMODORE 64:
- Utilidades - Lenguajes
- Sis. Operativos - Aplicaciones
- Juegos (nacionales e importación)
(Amplio surtido de programas educativos para sus hijos)

- × CASSETE COMMODORE LLAME
- × INTERFACE RS 232 LLAME
- × PADDLES LLAME
- × FUNDA COMMODORE 64 LLAME

SIEMPRE NUEVAS IDEAS EN FIRST NEWS. Y LO RECIBO GRATUITAMENTE. INCREIBLE.

FABULOSO REGALO TIP POSTER

A todos los propietarios de un VIC 20 ó C 64, manden un sobre indicando: Tipo de Ordenador, anualidad y lugar donde lo compró, tipo de configuración de su equipo. Recibirán: un FABULOSO TIP POSTER COMMODORE de FIRST. Manden su carta a FIRST TIP POSTER. Y por favor, dennos tiempo a contestar todas sus cartas.



CINTAS

CINTA CASSETTE INFORMATICA F20 (1)	135 Pts
CINTA CASSETTE INFORMATICA F20 (5)	600 Pts
CINTA CASSETTE INFORMATICA F20 (10)	1.100 Pts
CINTA IMPRESORA EPSON 80	800 Pts
CINTA IMPRESORA STAR	180 Pts

COMPRE SIEMPRE EN FIRST

PAPEL ESPECIAL PARA VD.

2.500 Hojas de papel (bordes perfectos). Superblanco. Grueso. 80c	4.456 Ptas.
2.500 Recibos negociables standard	4.663 Ptas.
1.000 Etiquetas autoadhesivas 2 × lineas)	828 Ptas.

Todos los envios de papel son a portes debidos.

```
1 INPUT"BO";A:POKE53280,A:INPUT"BA";B:POKE53281,B:INPUT
  "T";C:POKE646,C:GOTO1
```



Magic Desk

Ud y MAGIC DESK, penetrarán en la nueva generación de software. Imaginese usando su ordenador para hacer archivos y editar letras personales, etc. sin aprender un simple comando. Los comandos son GRAFICOS. Mueva sólo la mano animada y realize su deseo, listo para funcionar.
MAGIC DESK en FIRST............. LLAME

First In Quality Software

SUPER JOYSTICK

COMPATIBLE CON COMMODORE 64 y VIC 20
SINCLAIR (precisa interf)
ATARI
SPECTRAVIDEO

2.200!!

LOS LIBROS DE FIRST

COMMODORE-64	
Curso de Introducción al BASIC parte I	3.000
Curso de Introducción al BASIC parte II	3.000
Guia de Referencia del Programador	3.000
COMMODORE 64 COMPUTING	1.900
VIC-20	
VIC 20 GUIA DEL USUARIO	2.300
LEARN TO USE THE VIC-20	1.600
VIC-20 EXPOSED	1.900




ENTREGA INMEDIATA A PROVINCIAS

VENTA POR CORREO

Mande su pedido. Pago Talón conformado o giro postal. Pedidos inferiores a 4.500 Pts. sume 180 Pts. gastos envio. Pedidos de libros sólo no sume ningun gasto de envio. Catálogo mande 200 Pts en sellos. Pedidos OFERTA DEL MES sume 180 Pts. gastos envio (salvo que pida otros articulos que no sean de oferta)
La presente lista de precios es susceptible de ser modificada sin aviso previo.



pulséis una tecla y además situará un punto negro sobre la tecla del piano que está sonando en ese momento. De esta forma vosotros sois los que imprimís el ritmo que queráis. Si pulsáis 2, el ordenador hará sonar las notas de esta melodía a su ritmo normal a la vez que también nos indica sobre el teclado del piano la nota que suena en cada momento y suena un acompañamiento (cosa que no ocurre en la opción 1). He de advertiros, aunque ya lo comprobaréis, que aunque en pantalla aparece el teclado de un piano, el sonido que se escucha es de guitarra española, así como el de acompañamiento. Y dicho esto paso a explicaros el programa:

Líneas:

- 5-45: Presentación del título y autor (autor del programa, no de la melodía. Qué más quisiera yo). La línea 35 podéis modificarla a vuestro gusto siempre que nombre y apellidos no ocupen más de 13 caracteres ya que si no se saldrían del recuadro.
- 55-410: Como la sentencia REM indica se trata de dibujar en la pantalla el teclado de un piano. Hubiera sido más fácil usando PRINT, pero queda mucho más bonito hacerlo con POKE ya que el efecto que produce en pantalla da una impresión de sofisticación. La inicialización de la variable numérica A=54272 en la línea 10 resulta muy útil ya que (aunque creo que todos lo sabéis) 54272 es la diferencia entre una posición de la memoria de pantalla y la misma posición de la memoria de color así como la primera posición de memoria de los registros de sonido.
- 450-555: Ajustes sonoros de las voces 1 y 3 que son las que vamos a utilizar. La 1 para la melodía normal y la 3 para el acompañamiento. Una vez realizados estos ajustes, el sonido obtenido es el de guitarra española.
- 600: Se encarga de tomar los 90 valores de los 90 datas correspondiente a las 90 notas que componen la melodía. Cada dato esta compuesto de 10 números. Los 3 primeros son la alta frecuencia de cada nota. Los 3 siguientes son la baja frecuencia de cada nota. Los 2 siguientes son la duración de cada nota. Cuando una nota empieza a sonar, el valor de TI es igual a 0. Mientras su valor sea menor que estos 2 números, la nota seguirá sonando. Los 2 números restantes indican el lugar del piano en el que deberá aparecer el punto que indicará qué nota se está tocando en ese momento. A estos 2 números le sumaremos luego 1.864 para que el punto se sitúe en la posición de memoria de pantalla correspondiente con POKE.
- 610-640: Se encargan de que elijamos una de las 2 opciones anteriormente mencionadas.
- 700-750: Aquí es cuando empieza a sonar la melodía.
- 700: Inicializa el bucle que hará sonar las 90 notas y pone el reloj a 0 al principio de cada nota.
- 704: Calcula el lugar en el que aparecerá el punto sobre el piano.
- 705: Hace aparecer el punto sobre el piano.
- 710: Toma la alta frecuencia de cada nota.
- 720: Toma la baja frecuencia de cada nota, y verifica si hemos pulsado la opción 1 ó la 2.
- 725: Si hemos pulsado la opción 1 el ordenador espera a que pulsemos una tecla. Mientras no lo hagamos, la nota seguirá sonando.
- 727: Si hemos pulsado la opción 1 saltará a la línea 745.
- 730: Si hemos pulsado la opción 2, el ordenador hará sonar la nota mientras TI no sea igual al valor de los números 7 y 8 de cada dato contenido en A$. A la vez enviará la ejecución del programa a la subrutina 3000 que se encarga del acompañamiento.
- 745: Pone la envolvente a 0 para que deje de sonar la nota. También borra el punto que había sobre el teclado del piano.
- 750: Y el bucle se repite para continuar con otra nota.
- 800-2070: Datos que contienen la información antes mencionada. Estos datos se almacenan en una matriz A$ (j) de 90 elementos y mediante funciones de manipulación de cadenas como habéis visto en las líneas 704 a 730 descompone cada dato de 10 números en: 1 de 3 números, otro de 3 números, 1 de 2 números y otro de 2 números.
- 3000-3020: Subrutina de acompañamiento.

Y ya nada más. Aunque reconozco que es un palo lo de copiar todos los datas, pensad que más me ha costado a mí encontrar los valores y sinceramente creo que merece la pena. Recibid un saludo.

```
5 POKE53280,1:POKE53281,1
10 PRINT"[YEL][CLR][5CRSRD]"TAB(7)"[BLK]
EL[SPC]CONCIERTO[SPC]DE[SPC]ARANJUEZ
":A=54272
15 POKE1190,112:POKE55462,0:FORL=119
1TO1214:POKEL,64:POKEL+A,0:NEXT
20 POKEL,110:POKEL+A,0:POKE1255,66:P
OKE55527,0:POKE1295,125:POKE55567,0
25 FORL=1294TO1271STEP-1:POKEL,64:PO
KEL+A,0:NEXT:POKEL,109:POKEL+A,0
30 POKE1230,66:POKE55502,0
35 PRINT"[HOM][8CRSRD][10CRSRR]POR[SPC]
JOSE[SPC]L.[SPC]MUN[CRSRL][CRSRU]-[CRSRD]
OZ"
40 PRINT"[3CRSRD][SPC][SHIFTB]D[SHIFTB]
E[SHIFTB]F[SHIFTB]G[SHIFTB]A[SHIFTB]
B[SHIFTB]C[SHIFTB]D[SHIFTB]E[SHIFTB]
F[SHIFTB]G[SHIFTB]A[SHIFTB]B[SHIFTB]
C[SHIFTB]D[SHIFTB]E[SHIFTB]F[SHIFTB]
G[SHIFTB]A[SHIFTB]"
45 PRINT"[CRSRU][SPC][SHIFTB]2[SHIFTB]
2[SHIFTB]2[SHIFTB]2[SHIFTB]2[SHIFTB]
2[SHIFTB]3[SHIFTB]3[SHIFTB]3[SHIFTB]
3[SHIFTB]3[SHIFTB]3[SHIFTB]3[SHIFTB]
4[SHIFTB]4[SHIFTB]4[SHIFTB]4[SHIFTB]
4[SHIFTB]4[SHIFTB]"
55 REM DIBUJO PIANO CON POKES*******
***********************************
********
```



```
60 FORL=1585TO1623:POKEL,121:POKEL+A
,0:FORC=1TO8:NEXT:NEXT
65 L=1625:GOSUB400:L=1627:GOSUB400:F
ORL=1631TO1635STEP2:GOSUB400:NEXT:L=
1639
70 GOSUB400:L=1641:GOSUB400:FORL=164
5TO1649STEP2:GOSUB400:NEXT:L=1653:GO
SUB400
75 L=1655:GOSUB400:FORL=1659TO1663ST
EP2
80 GOSUB400:NEXT
85 FORL=1629TO1657STEP14:GOSUB410:NE
XT:L=1637:GOSUB410:L=1651:GOSUB410
90 POKE1945,109:POKE1945+A,0:POKE198
3,125:POKE1983+A,0
95 FORL=1946TO1981STEP2:POKEL,64:POK
EL+A,0:POKEL+1,113:POKEL+1+A,0
97 FORC=1TO20:NEXT:NEXT
100 POKEL,64:POKEL+A,0
200 PRINT"[9CRSRD][3SPC]OCTAVA-2[2SPC]
[SHIFTB][2SPC]OCTAVA-3[3SPC][SHIFTB]
[2SPC]OCTAVA-4[HOM]":GOTO450
400 FORK=LTOL+120STEP40:POKEK,160:PO
KEK+A,0:FORC=1TO8:NEXT:NEXT
402 FORM=KTOK+120STEP40:POKEM,93
405 POKEM+A,0:FORH=1TO8:NEXT: NEXT:R
ETURN
410 FORK=LTOL+280STEP40:POKEK,93:POK
EK+A,0:FORC=1TO8:NEXT:NEXT:RETURN
450 REM AJUSTES SONOROS VOCES 1 Y 3*
************************************
********
500 FORL=A TO A+23:POKEL,0:NEXT
510 POKEA+24 ,15:REM VOLUMEN
520 POKEA+5  ,25:REM ATAQUE/DECAE VO
Z 1
525 POKEA+19 ,25:REM ATAQUE/DECAE VO
Z 3
530 POKEA+6  ,24:REM SOSTEN/ESCAPE V
OZ 1
535 POKEA+20 ,27:REM SOSTEN/ESCAPE V
OZ 3
540 POKEA+3  ,0  :REM PULSO ALTO  VO
Z 1
545 POKEA+17 ,0  :REM PULSO ALTO  VO
Z 3
550 POKEA+2  ,255:REM PULSO BAJO  VO
Z 1
555 POKEA+16 ,255:REM PULSO BAJO  VO
Z 3
600 DIMA$(89):FORJ=0TO89:READ A$(J):
NEXT
610 PRINT"[HOM][CRSRD]1.-[SPC]MANUAL
":PRINT"2.-[SPC]AUTOMATICO[CRSRU]"
625 GETOP$:IFOP$="1"THENPRINT"[HOM][CRSRD]
[RVSON]1.-[RVSOFF]":GOTO700
630 IFOP$="2"THENPRINT"[HOM][2CRSRD]
[RVSON]2.-[RVSOFF]":GOTO700
640 GOTO625
700 FOR J=0TO89:TI$="000000"
704 C=1864+VAL(RIGHT$(A$(J),2))
705 POKEC,81 :POKEC+A,0:POKE54276,65
710 POKE A+1,VAL(LEFT$(A$(J),3))
720 POKE A,VAL(MID$(A$(J),4,3)):IFOP
$="2"THEN730
725 GETT$:IFT$=""THEN725
727 GOTO745
730 IFTI<VAL(MID$(A$(J),7,2))THENGOS
UB3000:GOTO 730
745 POKEA+4,0:FORB=1TO10:NEXT:POKEC,
32
750 NEXTJ:END
800 REM: ALTA FRECUENCIA,BAJA FRECUE
NCIA,DURACION DE CADA NOTA Y POKE EN
 LA
810 REM:MEMORIA DE PANTALLA PARA IND
ICAR LA NOTA QUE SE ESTA TOCANDO****
********
1000 DATA 0102051318,0091591616,0102
056018,0102051418,0111141520
1010 DATA 0122163122,0122161922,0122
162522,0111143220,0102051318
1020 DATA 0091591616,0102056818,0102
051118,0111141220,0122161622
1030 DATA 0141074824,0160474226,0122
162622,0111142020,0111142020
1040 DATA 0102052118,0102051518,0091
591316,0081471614,0091595916
1050 DATA 0102051618,0081471314,0080
231412,0081474614,0091591716
1060 DATA 0080231312,0070531410,0080
237212,0080231012,0081471514
1070 DATA 0091595816,0081471214,0080
231512,0080238112,0070535710
1080 DATA 0070531310,0080231212,0081
471414,0091592816,0102051218
1090 DATA 0111141520,0122161522,0141
071224,0122161322,0141074624
2000 DATA 0141071824,0160472126,0170
372028,0170372028,0170372128
2010 DATA 0160473626,0122161322,0111
141420,0122166622,0160471426
2020 DATA 0160471226,0141071324,0141
072224,0122162922,0111141120
2030 DATA 0122161122,0141074524,0122
160922,0111141020,0102051118
2040 DATA 0111144520,0102051018,0091
590916,0081471314,0091596016
2050 DATA 0081471714,0091591716,0102
051718,0111141720,0122161722
2060 DATA 0111141920,0102057618,0070
532010,0080231512,0081471314
2070 DATA 0091595616,0081471614,0080
231612,0080239912,0070539910
3000 REM SUBRUTINA ACOMPANAMIENTO **
****
3005 D=D+1:IFD<VAL(MID$(A$(J),7,2))+
15THENRETURN
3010 POKEA+18,65:POKEA+15,14:POKEA+1
4,107:FORH=1TO80 :NEXT:D=0
3020 POKEA+18,64:RETURN
```



¡¡¡Atención!!!

Alfonso Ramírez Ramírez
David Caballero
Daniel González Alonso
S. Castell Martorell

Enviadnos vuestras direcciones para que podamos devolveros las cintas que nos enviásteis.

VIC-20 + 3K expansión

Batalla

Ricardo David Alba Atencia
c/ Sumatra, Q-2
El Atabal (MALAGA)

Este juego simula una batalla entre dos jugadores, cada uno de ellos dispone de 23 ejércitos y otros tantos cañones, además de una bandera.

Cada jugador tiene una pantalla, de la que puede salir para enfrentarse con su oponente a través de uno de los tres puentes situados en el lado opuesto de la posición inicial de sus fichas.

Las fichas aparecen alineadas una debajo de la otra, a excepción de la bandera que está aislada.

Los cañones al igual que las banderas no se pueden mover. Los ejércitos pueden avanzar o retroceder un máximo de dos espacios en las direcciones del plano.

Si se desea mover una ficha habrá que posicionar el cursor sobre la ficha a mover mediante las teclas de movimiento de cursor, a continuación se pulsará la barra espaciadora con lo que la ficha quedará seleccionada para su movimiento. A continuación se volverá a mover el cursor hasta la posición de la pantalla en la que deseemos poner nuestra ficha y pulsaremos la tecla "@" con lo que la ficha quedará movida automáticamente.

Si queremos pasar una ficha de una pantalla a otra habrá que posicionarla antes al final del puente, pudiendo colocarse al turno siguiente en el otro lado del puente.

Cuando no se deseen mover todas la fichas, se moverán las que se deseen y a continuación se pulsará la tecla "=" con lo que el turno pasará al otro jugador apareciendo su pantalla.

El programa permite visualizar la pantalla deseada con solo la pulsación de la tecla "←" que visualizará la primera pantalla o la tecla "-" que visualizará la segunda pantalla.

A su vez al pulsar la tecla HOME el cursor se situará en la esquina superior derecha, al pulsar la tecla "£" el cursor se situará en la esquina superior izquierda, al pulsar la tecla "↑" se colocará en la esquina inferior derecha, y al pulsar la tecla "*" irá a la esquina inferior izquierda.

Si colocamos una ficha a una distancia de uno, dos o tres espacios de un tanque este dispara contra nosotros, y si obtiene una tirada inferior a cuatro nos habrá destruido, desapareciendo nuestra ficha del terreno de juego.

Un tanque será destruido al posicionar una de nuestras fichas sobre el tanque que deseemos destruir.

Dos fichas entraran en combate cuando estén juntas, resultando ganadora la ficha que consiga una puntuación mayor que la otra; en caso de empate, la tirada se repetirá.

Cada vez que el dado o los dados aparezcan, el programa se parará hasta que se pulse una tecla cualquiera.

El juego acaba cuando una ficha enemiga se posiciona sobre la bandera del otro jugador resultando este último perdedor de la batalla.

El juego corre en el VIC-20, pero necesita una ampliación de 3K de RAM. Si se dispone del SUPEREXPANDER, habrá que anular la influencia de dicho cartucho en la memoria ROM del ordenador, ya que interfiere en los gráficos del programa destruyéndolos por completo. Esto se consigue con el siguiente programa que debe ser ejecutado antes de cargar el programa BATALLA.GRAF. El programa es el siguiente:

```
POKE 256,120:POKE257,76:POKE258,47:POKE259,253:SYS256
```

El programa está dividido en dos:
—BATALLA.GRAF que crea los gráficos;
—BATALLA.PROG el juego propiamente dicho.

La segunda parte se divide en:

Líneas:

1-1000:	Inicialización de variables.
1020-1097:	Introducción de órdenes.
1097-1109:	Controla el cambio de turno.
3000-3040:	Cede el turno al otro jugador.
4000-4890:	Diversos casos de combate y simula dado.
5000-5100:	Paso de cursor a otra pantalla.
6000-6130:	Dibuja campo y fichas.
6900-7003:	Conserva el lugar pisado por el cursor.
8000-8080:	Elige ficha a mover.
8400-8640:	Mueve ficha y controla la posible batalla.
8900-9070:	Mueve el cursor a esquinas.
1000:	Mira el carácter que hay debajo del cursor.
11130-11268:	Mensaje de fin.

LISTADO 1

```
1 POKE36879,8
10 PRINT"[CLR][WHT][3CRSRD][2CRSRR][3SHIFT*]
[SHIFTI][SPC][SHIFTU][2SHIFT*][SHIFTI]
[SPC][SHIFTU][2SHIFT*][SHIFTI][SPC][SHIFTU]
[2SHIFT*][SHIFTI]
20 PRINT"[2CRSRR][SHIFT-][2SPC][SHIFT-]
[SPC][SHIFT-][2SPC][SHIFT-][2SPC][COMMM]
[COMMG][2SPC][SHIFT-][2SPC][SHIFT-]
30 PRINT"[2CRSRR][SHIFT-][2SPC][SHIFT-]
[SPC][SHIFT-][2SPC][SHIFT-][2SPC][COMMM]
[COMMG][2SPC][SHIFT-][2SPC][SHIFT-]
40 PRINT"[2CRSRR][3COMM@][SHIFTK][SPC]
[SHIFTJ][2SHIFT*][SHIFTK][2SPC][COMMM]
[COMMG][2SPC][SHIFTJ][2SHIFT*][SHIFTK]

50 PRINT"[2CRSRR][3COMMT][SHIFTI][SPC]
[SHIFTU][2COMMT][SHIFTI][2SPC][COMMM]
[COMMG][2SPC][SHIFTU][2COMMT][SHIFTI]

60 PRINT"[2CRSRR][SHIFT-][2SPC][SHIFT-]
[SPC][SHIFT-][2SPC][SHIFT-][2SPC][COMMM]
[COMMG][2SPC][SHIFT-][2SPC][SHIFT-]
70 PRINT"[2CRSRR][SHIFT-][2SPC][SHIFT-]
[SPC][SHIFT-][2SPC][SHIFT-][2SPC][COMMM]
[COMMG][2SPC][SHIFT-][2SPC][SHIFT-]
80 PRINT"[2CRSRR][3SHIFT*][SHIFTK][SPC]
[SHIFTK][2SPC][SHIFTJ][2SPC][COMMM][COMMG]
[2SPC][SHIFTK][2SPC][SHIFTJ]
```



SLOGAN

¿Utiliza sólo un

Utilizar un COMMODORE 64 únicamente para jugar, es como pedirle a Albert Einstein que nos resuelva la raíz cuadrada de cuatro.

El cerebro del ordenador apenas funciona.

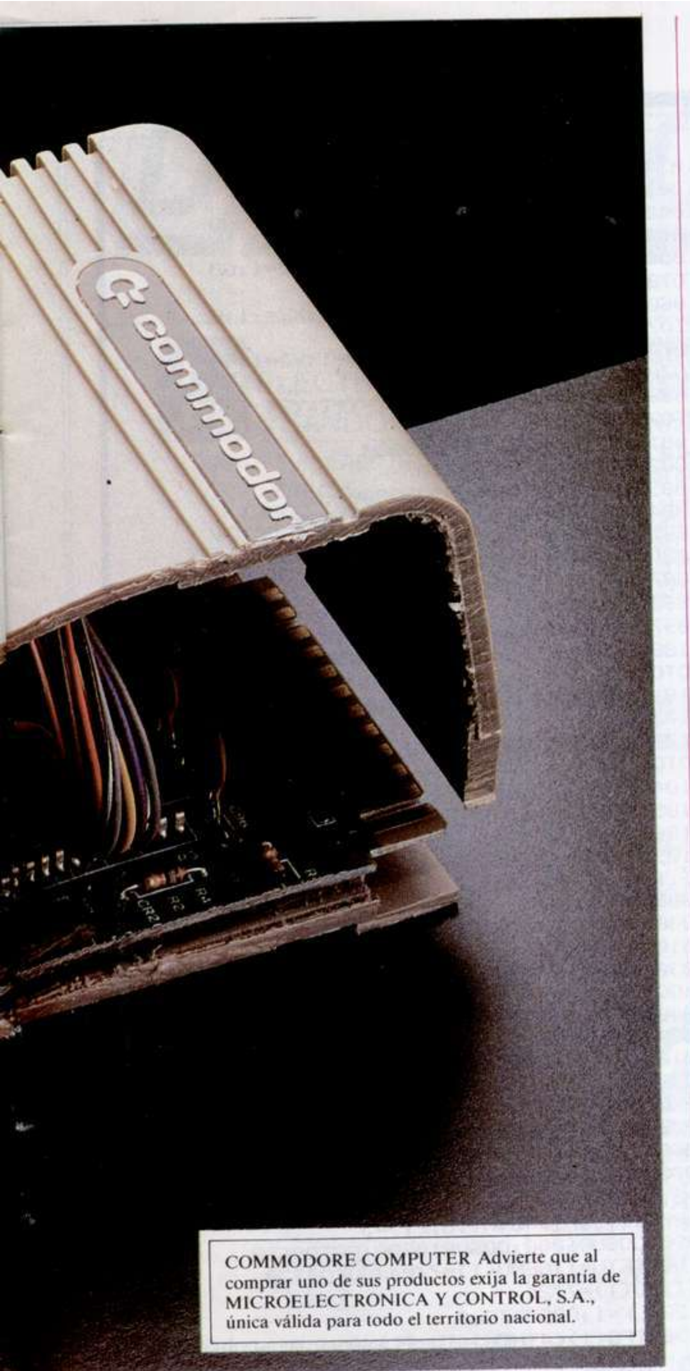
Para aprovecharlo al máximo, lo que Vd. necesita son programas interesantes (archivo de documentos, educación interactiva, estimulantes juegos o proceso de textos) y lógicamente los periféricos adecuados:

— La unidad de disco Commodore, un almacenaje de datos muy rápido, fiable y de gran capacidad.

— La unidad de cassette Commodore, una forma económica de introducir y almacenar programas.

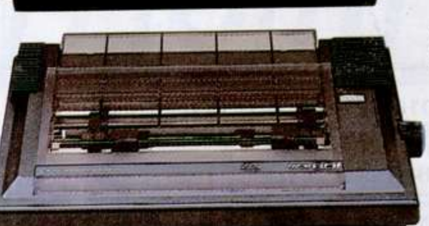
— El monitor en color Commodore, para

MICROELECTRONICA Y CONTROL c/ Valencia, 49-53. 08015 Barcelona. c/ Princesa, 47, 3.° G. 28008 Madrid.
UNICO REPRESENTANTE DE COMMODORE EN ESPAÑA

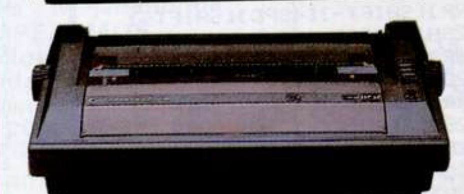




COMMODORE MPS 801
Impresora de matriz



COMMODORE MCS 801
Impresora cuatro colores



COMMODORE DPS 1101
Impresora de margarita



COMMODORE 1520
Printer plotter cuatro colores

COMMODORE 1541
Unidad de disco 170 K



COMMODORE 1530
Unidad de cassette



COMMODORE 1701
Monitor con color y sonido

JOYSTICKS
PADDLES
SOFTWARE Y LIBROS

COMMODORE COMPUTER Advierte que al comprar uno de sus productos exija la garantía de MICROELECTRONICA Y CONTROL, S.A., única válida para todo el territorio nacional.

10% de su cerebro?

lograr que los textos y gráficos sean más claros y vivos que en un televisor.

— 3 impresoras y un printer-plotter, para copias en papel, en blanco y negro o color.

— Los joystick y paddles, para conseguir que los juegos sean más excitantes.

Sea exigente. Si su cerebro, el de su COMMODORE 64, puede darle 100, no se conforme con 10.



la emoción del futuro



```
90 PRINT"[3CRSRD][4CRSRR][SHIFT*][SHIFTI]
[3SPC][SHIFT*][SHIFTI][3SPC][SHIFTU]
[2SHIFT*][SHIFTI]
100 PRINT"[4CRSRR][SHIFT-][4SPC][SHIFT-]
[4SPC][SHIFT-][2SPC][SHIFT-]
110 PRINT"[4CRSRR][SHIFT-][4SPC][SHIFT-]
[4SPC][SHIFT-][2SPC][SHIFT-]
120 PRINT"[4CRSRR][SHIFT-][4SPC][SHIFT-]
[4SPC][SHIFTJ][2SHIFT*][SHIFTK]
130 PRINT"[4CRSRR][SHIFT-][4SPC][SHIFT-]
[4SPC][SHIFTU][2COMMT][SHIFTI]
140 PRINT"[4CRSRR][SHIFT-][4SPC][SHIFT-]
[4SPC][SHIFT-][2SPC][SHIFT-]
150 PRINT"[4CRSRR][SHIFT-][4SPC][SHIFT-]
[4SPC][SHIFT-][2SPC][SHIFT-]
160 PRINT"[4CRSRR][SHIFTK][SHIFT*][SHIFTK]
[2SPC][SHIFTK][SHIFT*][SHIFTK][2SPC]
[SHIFTK][2SPC][SHIFTJ]
170 PRINT"[2CRSRD][3CRSRR][PUR]POR[SPC]
RICARDO[SPC]ALBA
980 G1=32768:G2=7168
990 FORI=0TO512:K=PEEK(G1+I):POKE(G2
+I),K:NEXTI
1000 DATA0,126,189,219,231,231,219,1
89,126
1002 DATA1,195,189,126,34,34,126,189
,195
1004 DATA8,0,238,238,0,0,187,187,0
1040 DATA3,255,255,255,255,255,255,2
55,255
1042 DATA6,0,224,192,224,240,224,192
,224
1044 DATA7,0,7,3,7,15,7,3,7
1046 DATA2,0,254,238,238,130,238,238
,254
1098 DATA49,255,255,255,255,239,255,
255,255
1100 DATA50,255,255,191,255,255,255,
251,255
1102 DATA51,255,255,191,255,239,255,
251,255
1104 DATA52,255,255,187,255,255,255,
187,255
1106 DATA53,255,255,187,255,239,255,
187,255
1108 DATA54,255,255,171,255,255,255,
171,255
1510 FORI=0TO12:READC:FORJ=0TO7:READ
K:POKEG2+(C*8)+J,K:NEXTJ:NEXTI
1600 POKE52,28:POKE56,28
```

LISTADO 2

```
1 POKE36879,8:A$(1)="[CRSRU][BLU][17CRSRR]
CCCCC":CH(1)=8180:X=12:Y=X:Y1=X:X1=X
:POKE36878,15
2 DIMJ(2,22),C(2,22),M(2,22):CH(2)=8
163:B$(1)="[CRSRU][RED][17CRSRR]HHHH
H":S=7680
5 A$(2)="[BLU]CCCCC":AC=32
411 POKE36869,255
415 CO=30720
417 B$(2)="[RED]HHHHH
420 FORI=0TO22:J(1,I)=S+2+I*22:J(2,I
)=-1*(J(1,I)+17):C(1,I)=J(1,I)+2:C(2
,I)=J(2,I)+2
421 NEXT:TU=1
1000 T=1:D=7701:GOSUB6000
1020 GETT$
1030 IFT$="←"THENT=1:GOSUB6000:GOSUB
10000
1040 IFT$="-"THENT=2:GOSUB6000:GOSUB
10000
1050 IFT$="[CRSRU]"THENX=X-1:X2=1:GO
TO7000
1060 IFT$="[CRSRD]"THENX=X+1:X2=-1:G
OTO7000
1070 IFT$="[CRSRL]"THENY=Y-1:Y2=1:GO
TO7000
1080 IFT$="[CRSRR]"THENY=Y+1:Y2=-1:G
OTO7000
1081 IFT$="[SPC]"THEN8000
1082 IFT$="@"THEN8500
1083 IFT$="£"THENGOSUB9000
1084 IFT$="[HOM]"THENGOSUB9020
1085 IFT$="*"THENGOSUB9040
1086 IFT$="↑"THENGOSUB9060
1087 IFT$="="THEN3000
1088 GOSUB5000
1097 POKES+Y+X*22,35
1100 FORI=0TO22:IFM(TU,I)=3THENNEXT:
GOTO1103
1101 IFM(TU,I)<>2THEN1111
1102 NEXT
1103 FORI=0TO22:IFM(TU,I)=3THENNEXT:
GOTO1105
1104 M(TU,I)=0:NEXT
1105 IFTU=2THENTU=1:GOTO1109
1106 TU=2
1109 T=TU:GOSUB6000:GOSUB10000
1111 POKES+Y+X*22,AC
2000 GOTO1020
3000 REM TERMINAR DE
3010 REM MOVER.
3030 FORRE=0TO22:IFM(TU,RE)=3THENNEX
T:GOTO1020
3040 M(TU,RE)=2:NEXT:GOTO1020
4000 REM BATALLA.
4010 IFT=1THEND(1)=7720:D(2)=7722:GO
TO4030
4020 D(1)=7703:D(2)=7705
4030 LK=2:GOSUB4800
4060 IFQ(1)=Q(2)THEN4030
4070 IFQ(1)>Q(2)THENJ(T2,I)=0:M(T2,I
)=3:GOTO4090
4080 J(TU,FI)=0:M(TU,FI)=3
4090 GOSUB6000:GOSUB10000:GOTO8620
4100 IFTU=1THEND(1)=7704:GOTO4120
4111 D(1)=7721
4120 LK=1:GOSUB4800
4130 IFQ(1)<4THENJ(TU,FI)=0:M(TU,FI)
=3
4140 SC=1:GOSUB6000:GOSUB10000:GOTO8
620
4800 FORTD=1TOLK
4810 FORG=0TO10:Q=INT(RND(1)*6)+1:PO
KE36876,128+Q:POKED(TD),48+Q:POKED(T
D)+CO,1
4820 GETT$:IFT$=""THENNEXT
4830 POKE36876,0:Q(TD)=INT(RND(1)*6)
+1:POKED(TD),48+Q(TD):NEXT
4835 GETT$:IFT$=""THEN4835
4840 RETURN
4850 IFTU=1THENT2=2:GOTO4855
4851 T2=1
4855 FORI=0TO22:IFABS(C(T2,I))=S+Y+X
*22THEN4870
4866 NEXT
4870 C(T2,I)=0
4890 GOSUB6000:GOSUB10000:RETURN
4900 IF-M=P-21THEN4000
4910 IF-M=P+1THEN4000
4920 IF-M=P-43THEN4000
4940 GOTO8639
5000 REM PASO A OTRA
```

DIRECTORIO



DISTRIBUIDORES DE:
ORIC-1
CASIO FP200
ROCKWELL-AIM-65
VIDEO GENIE-EG-2000
CASIO FX-9000P
SINCLAIR SPECTRUM
OSBORNE 1
DRAGON-32
NEW BRAIN
EPSON HX-20

C/SANDOVAL, 3, 4, 6 MADRID 28010
TELEFONOS 445 18 33/18 70
TELEX 47784 SAVL-E

casa de software s.a.

c/ aragón, 272, 8º, 6.ª
tel. 215 69 52 - 08007 barcelona

- Software profesional para C-64
- Con distribución productos DIGITAL RESEARCH



- Ordenadores personales Hard y Soft.
- Cursos de Basic.

Oficinas: **RENOVACION EN MARCHA, S.A.**
c/. Espronceda, 34 - 2º int. - MADRID-3
Teléfono **(91) 441 24 78**
Tienda: **REM SHOP 1**
c/. Galileo, 4 - MADRID-15
Teléfono **(91) 445 28 08**

COMPUTERWORLD ESPAÑA

es miembro de CW Communications/Inc., el grupo editorial más grande y de publicaciones relacionadas con los ordenadores en 18 países, dando como resultado la cantidad de nueve millones de lectores que leen una o más de nuestras publicaciones.

¡¡¡Suscríbete!!!
Tel. (91) 231 23 88/95

CENTRO DE INFORMATICA

Las Rozas - Majadahonda
EMPEZAMOS
Cursillos en BASIC
cada 15 días
Directamente en ordenadores
VIC-20, COMMODORE 64
SPECTRAVIDEO
Teléfono: 637 31 51

FERRE-MORET S.A.

C/ Tusset, 8 - Entresuelo 2.ª
Tel.: 218 02 93
08006 BARCELONA



- ORDENADORES PERSONALES
- ACCESORIOS INFORMATICA
- COMPONENTES ELECTRONICOS
- TELECOMUNICACIONES

Paseo de Gracia 126-130
Tel. 237 11 82. 08008 BARCELONA*

MICROS GARDEN SA
- ORDENADORES PERSONALES -

- SINCLAIR SPECTRUM Y PLUS
- COMMODORE 64
- ATARI 800 XL
- ORIC ATMOS
- DRAGON 32 y 64

- PERIFERICOS Y ACCESORIOS
- SOFTWARE PARA TODAS LAS MARCAS
- CURSOS DE BASIC PARA PRINCIPIANTES (PRACTICAS CON ORDENADOR)
- GRAN SURTIDO EN LIBROS Y REVISTAS

Francisco Silvela, 19
Teléfono 401 07 27. 28028 MADRID



HACEMOS FACIL LA INFORMATICA

• SINCLAIR • SPECTRAVIDEO
• COMMODORE • DRAGON
• AMSTRAD • APPLE
• SPERRY UNIVAC

Modesto Lafuente, 63 Telf. 253 94 54 28003 MADRID	Colombia, 39-41 Telf. 458 61 71 28016 MADRID
José Ortega y Gasset, 21 Telf. 411 28 50 28006 MADRID	Padre Damián, 18 Telf. 259 86 13 28036 MADRID
Fuencarral, 100 Telf. 221 23 62 28004 MADRID	Avda. Gaudí, 15 Telf. 256 19 14 08015 BARCELONA
Ezequiel González, 28 Telf. 43 68 65 40002 SEGOVIA	Stuart, 7 Telf. 891 70 36 ARANJUEZ (Madrid)

ORDENADORES

PERSONALES/PROFESIONALES

RED DE ORDENADORES PARA COLEGIOS.



MICROINFORMATICA

Equipos/Servicios
Centro de formación

General Ricardos, 15.
Tel. (974) 31 14 39. BARBASTRO (Huesca)

MicroSistemas

Todos los meses publicaremos programas para los Commodore.

¡¡Suscríbete!!

Tel. (91) 231 23 88/95



```
5010 REM PANTALLA.
5030 IFY>21ANDT=2THENY=21
5040 IFY>21ANDT=1THENY=0:T=2:GOTO510
0
5050 IFY<0ANDT=2THENY=21:T=1:GOTO510
0
5060 IFY<0ANDT=1THENY=0
5090 RETURN
5100 GOSUB6000:GOSUB10000:RETURN
6000 REM DIBUJA CAMPO
6001 REM Y FICHAS.
6020 PRINT"[CLR]";
6030 PRINT"[HOM]";:FORH=1TO3:FORI=1T
O5:PRINTA$(T):NEXT
6040 FORI=1TO2:PRINTB$(T):NEXT:NEXT
6050 PRINTA$(T)"[HOM]
6060 FORI=1TO5:POKECH(T)+I,3:POKECH(
T)+I+CO,6:NEXT
6065 IFT=2THEN6100
6070 POKE7900,2:FORI=0TO22:IFJ(1,I)>
0THENPOKEJ(1,I),0:POKEJ(1,I)+CO,7
6080 IFC(1,I)<>0THENPOKEC(1,I),6:POK
EC(1,I)+CO,5
6081 IFJ(2,I)>0THENPOKEJ(2,I),1:POKE
J(2,I)+CO,3
6090 NEXT:RETURN
6100 POKE7921,2:FORI=0TO22:IFJ(2,I)<
0THENPOKEABS(J(2,I)),1:POKEABS(J(2,I
))+CO,3
6110 IFC(2,I)<>0THENPOKEABS(C(2,I)),
7:POKEABS(C(2,I))+CO,4
6120 IFJ(1,I)<0THENPOKEABS(J(1,I)),0
:POKEABS(J(1,I))+CO,7
6130 NEXT:RETURN
6900 REM CONSERVAR
6901 REMLUGAR PISADO.
7000 A2=PEEK(S+Y+X*22)
7001 IFA2=3ORX<0ORX>22THENX=X+X2:Y=Y
+Y2:GOTO7004
7002 IFY=22ORY=-1THEN7004
7003 AC=A2
7004 POKE36876,200:POKE36876,0:X2=0:
Y2=0:GOTO1081
8000 REM ELEJIR FICHA
8010 REM A MOVER.
8012 IFT=2THEN8050
8020 FORFI=0TO22:IFJ(TU,FI)=S+Y+X*22
THEN8035
8030 NEXT:FI=-1:GOTO1020
8035 IFM(TU,FI)=0THENM(TU,FI)=1
8040 GOTO1020
8050 FORFI=0TO22:IF-1*(J(TU,FI))=S+Y
+X*22THEN8070
8060 NEXT:FI=-1:GOTO1020
8070 IFM(TU,FI)=0THENM(TU,FI)=1
8080 GOTO1020
8400 REM MOVER FICHA.
8500 IFFI=-1THEN1020
8501 IFM(TU,FI)=2ORM(TU,FI)=3ORM(TU,
FI)=0THEN1020
8509 Y3=(ABS(J(TU,FI)))-S:X3=0
8510 IFY3<22THEN8541
8520 Y3=Y3-22
8530 X3=X3+1
8540 GOTO8510
8541 IFY=0ANDT=2ANDTU=1ANDY3=21THENY
3=-1:Y=0:GOTO8560
8542 IFY=21ANDT=1ANDTU=2ANDY3=0THENY
=21:Y3=22:GOTO8560
8543 IFY=21ANDT=1ANDTU=1ANDY3=0THENY
=21:Y3=22:GOTO8560
8544 IFY=0ANDT=2ANDTU=2ANDY3=21THENY
3=-1:Y=0:GOTO8560
8545 IFY<2ANDTU=1ANDT=1THEN1020
8546 IFY>19ANDTU=2ANDT=2THEN1020
8547 QW=J(TU,FI)
8548 IFQW<0ANDT=1THEN1020
8549 IFQW>0ANDT=2THEN1020
8560 IFX3>X+2ORX3<X-2ORY3>Y+2ORY3<Y-
2THEN1020
8562 IFPEEK(S+Y+X*22)=2ANDTU<>TTHEN1
1000
8563 IFPEEK(S+Y+X*22)=6ANDTU=2ORPEEK
(S+Y+X*22)=7ANDTU=1THENGOSUB4850
8569 IFPEEK(S+Y+X*22)=8THEN8580
8570 IFPEEK(S+Y+X*22)<>32THEN1020
8580 POKEABS(J(TU,FI)),32
8590 POKES+Y+X*22,TU-1:POKECO+S+Y+X*
22,1:AC=TU-1:X3=0
8591 VA=1:IFT=2THENVA=-1
8592 IFTU=2THENT2=1:GOTO8600
8593 T2=2
8600 J(TU,FI)=S+Y+X*22:M(TU,FI)=2
8610 J(TU,FI)=J(TU,FI)*VA
8619 SC=0
8620 P=J(TU,FI):FORI=0TO22:M=J(T2,I)
:C=C(T2,I)
8630 IFP=M+1ORP=M-1ORP=M+22ORP=M-22O
RP=M+23ORP=M-23ORP=M-21ORP=M+21THEN4
010
8631 IFTU=TTHEN8635
8632 IFSC=1THEN8635
8633 IFP=C+1ORP=C+2ORP=C+3THEN4100
8635 IFY=21ANDT=1ORY=0ANDT=2THEN4900

8639 NEXT:GOTO1020
9000 IFT=2THENX=0:Y=5:GOTO10000
9010 X=0:Y=0:GOTO10000
9020 IFT=2THENX=0:Y=21:GOTO10000
9030 X=0:Y=16:GOTO10000
9040 IFT=2THENX=22:Y=5:GOTO10000
9048 X=22:Y=0:GOTO10000
9060 IFT=2THENX=22:Y=21:GOTO10000
9070 X=22:Y=16:GOTO10000
10000 AC=PEEK(S+Y+X*22):RETURN
11000 REMFIN.
11010 REM
11020 FORI=128TO255:POKE36876,I:NEXT
:FORI=255TO127STEP-1:POKE36876,I:NEX
T
11230 POKE36869,240:PRINT"[CLR][4CRSRR]
G[SPC]A[SPC]N[SPC]A[SPC]D[SPC]O[SPC]
R
11240 PRINT"[5CRSRD][CRSRR]JUGADOR:[SPC]
"TU
11250 PRINT"[3CRSRD][CRSRR]OTRA[SPC]
PARTIDA?
11260 GETT$:IFT$=""THEN11260
11268 IFT$="S"THENRUN
```



Programación basic para microcomputadoras

Autor: Luis Joyanes Aguilar
Nº pág.: 440.
P.V.P.: 1.490 Ptas.
Editorial Mc Graw Hill España
C/ Santa Beatriz, 4
Madrid
Teléf.: 91/433 87 77

Este libro va destinado tanto para aquellos que comienzan en el mundo de la informática, como para los que ya llevan algún tiempo y desean conocer un poco más a fondo las peculiaridades del basic en los microordenadores más populares del mercado nacional. El autor recomienda a los principiantes que lean detenidamente cada capítulo y practiquen sobre él, antes de pasar al siguiente.

En cada capítulo presenta una serie de ejercicios para que el lector los resuelva sobre un cuaderno o sobre el mismo ordenador, comprobando las aplicaciones del basic estudiadas en cada capítulo y verificando posteriormente los resultados con los que aparecen en el libro.

El capítulo cero del libro está dirigido a los que llegan ahora a la informática. Hace una "microhistoria" del desarrollo de los ordenadores desde su comienzo hasta nuestros días, y expone de modo sencillo los conceptos fundamentales: caracteres, datos, clasificación de las computadoras, hard y soft, el concepto de programa y la arquitectura y funcionamiento de un ordenador (CPU, ALU, entradas/salidas, RAM, ROM, impresoras, cintas, discos, etc.).

El capítulo uno enseña a manejar el ordenador, tanto en modo directo como en modo de programa, estudia las sentencias fundamentales (LET, PRINT, etc.) como editar, ejecutar, modificar y corregir un programa. También indica cómo guardar y cargar programas para volverlos a utilizar posteriormente.

El capítulo dos y siguientes ya explican más detalladamente y a un nivel superior la utilización de diagramas de flujo, variables, sentencias iniciales (input, print, rem), sentencias de control (goto, if then, for next, on goto), manejo de caracteres y funciones de cadenas (len, val, left$, right$, space$, etc.), funciones generales (max, min, int, etc.), trigonométricas, definidas por el usuario, entradas y salidas de datos (read/data/restore, spc, tab, pos, print at, print using).

A partir del capítulo 8, el libro sube su nivel, comienza el estudio de la utilización de listas y tablas, dimensionado de matrices (dim); option base, mat print y mat input. También estudia las subrutinas e instrucciones de llamada y retorno. En el capítulo 10 encontramos el estudio de los Peeks y los Pokes, call, usr, sys, memfre, ti$, y otras funciones especiales (var, wait, swap) y las órdenes de control y depuración de un programa (on error goto y trace entre otras).

Una vez terminado el estudio del basic y su utilización, en el capítulo 11 amplía las nociones que dio en el primer capítulo para detectar errores, modificar programas, numeración de líneas de los programas, etc., con una serie de detalles que si los hubiese intentado explicar anteriormente no los habrían entendido una gran parte de los lectores.

En definitiva este libro es un buen curso de basic o manual de consulta para todos, están bien desarrollados todos sus capítulos y son fácilmente comprensibles siempre que se tengan los conocimientos necesarios para cada uno de ellos. Por esto es importante, en caso de no estar seguro de saber algo, empezar el estudio desde los primeros capítulos, y quien ya tenga conocimientos puede pasar directamente a capítulos superiores.

Uno de los detalles que hacen más interesantes a este libro, es que no se limita a estudiar el Basic del VIC o del C-64, todas las explicaciones son válidas para el basic en general, por lo que sigue siendo útil aunque cambiemos de ordenador. Además, en sus apéndices, aparecen el juego de caracteres ASCII y las variantes utilizadas en los ordenadores personales más populares del mercado español (IBM/PC, PET/CBM, APPLE II, ZX-81, SPECTRUM, VIC-20/C-64) y el de las impresoras SHEIKOSA.

Otro de los apéndices detalla las características de las impresoras más utilizadas actualmente (EPSON y SHEIKOSA). El siguiente repite las típicas y útiles tablas de conversión decimal-hexadecimal (que nunca vienen mal), y por último un detalle interesantísimo; un apéndice describiendo los comandos y funciones de cada uno de los ordenadores que cité anteriormente. Esto nos permite poder adaptar a nuestro ordenador algunos programas que consideramos interesantes y que aparezcan publicados para otros ordenadores de diferentes marcas, ya que en la descripción de cada función podemos ver lo que hace e imaginar el modo en que podríamos simular la función de ese comando en nuestro ordenador con otro comando o conjunto de comandos equivalentes. Estos apéndices son la parte más útil de libro para mí, y no me considero un principiante en informática...

Commodore 64. Qué es, para qué sirve y cómo se usa

Autores: D. Ellershaw/P. Schofield
Traductor: Ramos Tapia.
Nº Pág.: 160.
P.V.P. 950 Ptas.
Editorial Noray
C/San Gervasio de Cassolas, 79
08022 BARCELONA

Una buena ayuda para principiantes

Esta obra va dirigida para aquellos usuarios del ordenador que acaban de comprarlo y se encuentran desconcertados. Quizás estas navidades los chicos pidieran a SS.MM. los Reyes Magos un ordenador "por que su amigo tenía uno...", o en el colegio lo están empezando a utilizar o, es más barato a la larga que los vídeo-juegos, y además papá quiere saber qué es "eso" e intentar llevar una pequeña contabilidad o fichero, etc. El caso es que muchas veces los usuarios se encuentran con un ordenador en su poder y no saben cómo usarlo ya que el manual es demasiado complicado y no aclara muchas dudas.



Para todos estos propietarios del Commodore 64 va destinado este libro, del que su autor dice en el primer capítulo (introducción): "Si encuentra usted el manual difícil de seguir, si ve que comprende solamente las primeras líneas y poco más, entonces ESTE ES SU LIBRO". Estamos totalmente de acuerdo con el autor en esta afirmación, ya que aunque parezca mentira hemos recibido varias decenas de llamadas de personas que han impuesto su dinero a plazo fijo en algún banco, les han dado a cambio el ordenador, y no sabían ni siquiera cómo conectarlo para utilizarlo.

El libro, que originalmente se titula "Very Basic Basic" (Basic muy básico), está bien traducido. Además una nota curiosa: cuesta menos la versión en español que la original inglesa, al menos el ejemplar que hemos encontrado a la venta en uno de los grandes almacenes de la capital.

Esta obra ha sido dividida en cuatro secciones, la primera de ella titulada "antes de programar", como podéis suponer da una serie de consejos, desde cómo conectar el aparato hasta algunos programas simples, pero su función principal es la de familiarizar al lector con los comandos basic más sencillos y cómo utilizarlos en los programas (LET, PRINT, etc).

En la segunda sección trata el manejo de las variables y las técnicas de programación desde un punto de vista elemental, claro y conciso. La utilización de las variables, bucles, contadores y bifurcaciones han sido ilustradas con numerosos ejemplos que afortunadamente no son sólo teóricos, sino que tienen aplicaciones prácticas, por ejemplo: en el capítulo de manejo de cadenas, para ilustrar el modo de utilizarlas, encontramos un "procesador de textos primitivo" que ocupa solamente seis líneas de programa.

La tercera sección está dedicada al manejo del grabador de cinta (datassette) y la impresora, son sólo dos cortos capítulos con una serie de instrucciones y consejos útiles para el manejo y conservación de estos dos dispositivos.

(Pasa a pág. 72)

Magia



La MAGIA son trucos, la MAGIA es divertida.

La MAGIA es hacer lo que nadie se ha atrevido y resulta ser la fuente más completa de información para la informática práctica.

La MAGIA es una sección mensual llena de consejos, trucos, de esto y aquello del mundo del software, hardware y aplicaciones, trucos descubiertos por los demás que hacen que la informática sea más fácil, más divertida o más animada.

MAGIA habla de ideas sencillas, programas de una sola línea, subrutinas útiles, hechos de informática poco conocidos y otras cosas de interés.

Los trucos de magia enviárnoslos comprobados, pues hay varios incorrectos. ¡Ah! y no nos mandéis trucos repetidos... ¡Listos!. Premios: Ver pág. 5.

VELOCIDAD EN LISTADOS

Hola Commodorianos. Esta es la primera vez que escribo a una revista, y os envío un pequeño truco.

¿Os molesta la lentitud con que va nuestro C.T.R.L.?, ¿cuando listando un programa lo presionáis no os molesta su lentitud? Pues ahí va el truco, apretando el C.T.R.L., apretar a la vez cualquiera de estas teclas, "C=", "RUN STOP", "1", "2", "Q". El listado pasará rápido y cuando soltéis la segunda tecla continuaréis con la lentitud de C.T.R.L. Y una pregunta, ¿por qué al encender mi 64 y poner PRINT FRE (8) me sale —26627 de memoria libre?

Alonso Soler Mellado
C/Miguel Reverter, 9-1º-3º
San Justo Devern (Barna)

NOTA: La respuesta es que cuando conectas el 64, este tiene más de 32K libres, e interpreta esta cantidad como negativa. Para saber cuál es la memoria verdadera, teclea:
PRINT 2116-fre(Ø).

UNA SOLA LINEA

Leyendo la sección de Magia de vuestra revista me ha animado a mandaros un par de PROGRAMAS DE UNA SOLA LINEA no tienen utilidad práctica pero, al menos el primero creo que es algo didáctico para aprender los POKES de activación de sonido a la vez que produce un efecto de pasos con eco, esta línea debe de introducirse usando las abreviaturas de los comandos que intervienen en ella:
SIN ABREVIAR:

```
1V=54296:fori=35TO0step-.3:pokeV, 1x2: poke V-2,235: poke V-1,
1x7: poke V-5,94: poke V-6,230: next: goto1
```

En cuanto a la segunda no precisa ni mucho menos de los 80 caracteres pero el efecto visual conseguido en la pantalla del monitor es divertido:

```
2 PRINT"[SHIFT-CLEAR]": FORI=1TO255:POKE53281,I: POKE
53280, I: NEXT: RUN
```

La he numerado con un dos esta segunda línea porque si las introducimos las dos en el CBM-64, y eliminamos el GOTO1 de la revista se produce un nuevo efecto, es más, si manejamos únicamente la primera línea y variamos el valor segundo de los POKES se pueden crear nuevas pautas de sonido.

Arcadio Segura Arnau
C/San Miguel, 61
La Cenia (Tarragona)

EL BUFFER DEL TECLADO

Básicamente, lo que hace es anular el buffer del teclado, que puede ser, en algunas ocasiones, muy engorroso. Voy a poneros un ejemplo "mu" clarito:

```
10 FOR N=Ø to 499
20 PRINT N
30 NEXT N
40 INPUT A$
```

¡¡Qué programas más bonitos me salen!!. No, en serio. Cuando corramos este programa, **mientras se esté ejecutando el bucle** (que se puede ver por dónde va gracias a la línea 20), nosotros teclearemos una palabra, por ejemplo "MI CASA", (por eso del E.T.) y cuando se termine el bucle, ¿qué pasa?, que nos sale el interrogante del INPUT, seguido de la palabra ¡¡MI CASA!!. Sobre todo, ¡¡Antes de que apareciese el interrogante!!.

Pues no quiero ni contaros lo que pasa si sustituimos la línea 40 por GET A$: IF A$=" " THEN 40.

Esto resulta verdaderamente incómodo cuando se tiene un programa activado por MENU y con sentencias GETs, como, por ejemplo, el programa de AGENDA que os mandé (y del que no sé qué vais a hacer con él, pobrecito).

Todo esto se puede evitar con un sencillo truco, que es ¡anular el teclado! con los POKEs 649.

Si ponemos estas nuevas líneas:

```
5 POKE 649, Ø
35 POKE 649, 1Ø
```

Se elimina ese molesto inconveniente. Probad ahora. Si queréis divertiros, cuando el contador del bucle vaya por el número 250, pulsad la tecla RUN/STOP.

Es para el Commodore 64.

Juan Rafael Oscar Martín y Mihalic
C/Guayadeque, 4-3º Izda.
35009 Las Palmas de G. Canaria.

DESATASCADOR

Os envío este comando DESATASCADOR por si os puede solucionar algún problema, para el Commodore 64.

Ocurre algunas veces trabajando con el ordenador que perdemos totalmente el control del mismo y la pantalla empieza a hacer cosas raras. Pulsamos entonces la tecla RESET (Commodore World nº 3. Pág. 38) muy recomendable para estos casos.

Ahora de lo que se trata es de recuperar el programa sin tener que cargarlo nuevamente cosa que será posible siempre que el atasco no lo haya destruido.

Para ello pulsar la siguiente **secuencia de teclas:**

(CLR)(SPC) 3 (RVS-ON) % % (RVS-OFF) " " (DEL) (RVS-ON) E (RVS-OFF) - (RVS-ON) % (RVS-OF) # (RVS-ON) E (RVS-OFF). (SHIFT + SPACE)

que nos dejarán estos signos en la pantalla:

D=DIRECTO

I=INVERSO......

D	D	I	I	D	I	D	I	D	I	D	D
	3	%	%	"	E	—	%	#	E	.	
SPC					GUION						SPC+SHIFT

"Sin hacer RETURN" bajamos una línea con la tecla de cursor (CRSRD) y escribimos la siguiente línea:

POKE 2050, 3: SYS 1024: POKE 45, PEEK (45) + 2: CLR (RETURN)

Si nos diera el mensaje: "ILLEGAL QUANTITY" hacer: POKE 46, PEEK (46) + 1: POKE 45, 2: CLR (RETURN).

José Ramón Lasa
Matxiategui, 34-4º
Bergara (Guipúzcoa)

NUMERO DE COLUMNAS

Esta es mi primera colaboración pero creo que puede ser muy interesante para los usuarios del VIC. Se trata de un truco de magia que aumenta o disminuye el número de columnas; puede tener desde 1 columna a 27, pero hay unos problemillas: uno de ellos es que cuando se pone un número de columnas mayor de 22 salen en la parte baja de la pantalla unos signos raros y el otro es que el cursor baja y sube de 22 en 22 y es difícil de situarle donde se desea, sin más paso a explicar el truco.

POKE 36882, X siendo X un número de 129 a 155

POKE 36864, Y siendo Y el número que mejor centre la pantalla.

Un ejemplo de 27 columnas:

```
10 POKE 36882, 155
20 POKE 36864, 10
30 PRINT "[CLR]"
```

Mario García Anibarro
C/Burgos, nº 8 - 1º C
Socio nº 588
Móstoles (Madrid)

YA TENEMOS "Ñ"

Os mando un pequeño secreto para que, a partir de ahora, podamos poner la letra "Ñ" como está en nuestro idioma y así no tener que hacer o poner las virguerías o sustitutos que habíamos tenido que utilizar. Veréis que se trata de, mediante los cursores, poner el siglo "-" (menos) encima de la letra "N". Espero que os guste y os sirva.

PRINT "NIN(CRSR ARRIBA) (CRSR IZQUIERDA) - (CRSR ABAJO) ITO"

(Imprimirá la palabra **NIÑITO**).

Salvador Lahoz Ros
C/ Federico García Lorca, 16-3º-3ª
08033 MADRID

VELOCIDAD DE LISTADO

Os mando esta colaboración para el VIC-20: El POKE37877,Ø, reduce la velocidad del listado de cualquier programa. También aumenta la velocidad de movimiento del cursor.

El POKE37877, 25Ø reduce la velocidad de movimiento del cursor.

C. Luis Cerón.
Plaza de España, 5, 2º
Almendralejo (BADAJOZ).

ACCESO AL MAPA DE MEMORIA

Escribo a esta sección para añadir respecto al artículo "Viaje al centro de tu Commodore", perteneciente al nº 8, un cierto truco para acceder al mapa de memoria.

Tecleando POKE 53272,4:

Fijémonos en la posición (12,6) (columna, fila) respectivamente; si pulsamos por ejemplo la tecla INST/DEL, vemos cómo aparecen consecutivamente casi toda la gama de caracteres posibles (probad con otras teclas). A su vez son interesantes de observar los POKE 53272, 8/50/65/80/96.

Si pulsamos SHIFT y COMMODORE, veremos ciertas cosas de "MAGIA".

Para retornar a la normalidad RUN/RESTORE.

Carmelo García del Rich
C/Llull, 169, bis
08005 BARCELONA.

FUEGOS ARTIFICIALES

Este es un pequeño "truco" o mejor dicho, un programa que simula unos fuegos artificiales. Sólo funcionará con el S. Expander, ya que utiliza funciones gráficas que el VIC standar no posee. Sin más os escribo el programa.

```
10 REM FUEGOS ARTIFICIALES
20 INPUT "(CLR) QUE MODO GRAFICO"; MG
30 IF MG<1ORMG>2 THEN 20
40 INPUT "(CLR) COORDENADA DE X";X
50 IF X<0 OR X>1023 THEN 40
60 GRAPHIC MG
70 CY=INT (RND(1)*1024)
80 CX=INT (RND(1)*1024)
90 CO=INT (RND(1)*8*MG)
100 REGION CO: DRAW 2,X,1023 TO CX,CY
110 GOTO 70.
```

Carlos Rojas López
C/Gustavo Gallardo, 7, 4º D
Telf.: (954) 61 41 24

SIMPLIFICANDO

Les envío una receta para el apartado de MAGIA.

Creo que será de gran utilidad, sirve para simplificar el teclear los comandos DATAS.

1.—Se debe introducir de modo directo la instrucción siguiente: POKE 650,128

2.—A continuación se debe escribir la línea 1000 como sigue: 1000 FORI=5TO 5 105 STEP5: PRINT;"DATA":NEXT

3.—Luego se ejecuta RUN 1000 con lo que la pantalla se llenará de números de línea acompañados del comando DATA.

4.—Se acciona la tecla HOME y se debe mantener apretada la tecla RETURN pulsada hasta el final.

A continuación vaya a la línea 1000, y sustituya los dos 5 que figuran en la misma por el último número aparecido en pantalla. Se vuelve a ejecutar RUN 1000 y así hasta terminar con las líneas de los comandos DATAS.

José María Fondevilla
Suscriptor 5130

El Misterio de los Pokes (IX)



Por Diego ROMERO

En este número vamos a ver algunas de las posiciones más importantes del mapa de memoria de los "hermanos mayores", es decir, los equipos Commodore de la gama alta. También podemos conocer la manera de pasar programas de un VIC a un PET 3000.

Los punteros que nosotros encontramos en el VIC y el C64 en las direcciones $2B, $2C, $2D, etc., son los que indican el principio y final de la memoria ocupada por programas Basic, principio y final de variables, etc. Todos ellos ya los he explicado en números anteriores, y para encontrar los equivalentes publicamos esta tabla con la dirección correspondiente del C-64 y su equivalente en el PET-CBM 3000.

VIC	PET
$2B-2C	$28-29
$2D-2E	$2A-2B
$2F-30	$2C-2D
$31-32	$2E-2F
$33-34	$30-31
$35-36	$32-33
$37-38	$34-35

Creo que estas son las posiciones más importantes a la hora de intercambiar programas, ya que la única dificultad que se nos presenta al cargar un programa de cinta de un VIC en un PET 3000 es que el VIC carga normalmente los programas en la primera posición de RAM que tenga libre, excepto que el indicador de carga o dirección secundaria utilizada al hacer save sea un 1, con lo que al hacer LOAD la carga se realiza en las mismas direcciones de donde salió el programa, igual que si hiciésemos LOAD" ",1,1.

Para poder intercambiar los programas en cinta con un PET debemos añadir una línea 0 REM en el PET al principio del programa, luego hacemos LOAD" ", 1 para cargar un programa grabado por un VIC, y luego hacemos POKE 1025, 1: POKE 1026,16. Con estos pokes alteramos el enlace de la primera línea Basic (el 0 REM) a la posición donde se encuentra la primera línea del programa del VIC, luego podemos hacer LIST y si queremos eliminamos la línea cero tecleando un cero y pulsando Return. La única condición para que esto funcione es que el programa del VIC 20 no contenga la línea cero y que sea para el VIC 20 sin ampliación. En caso de disponer de ampliación mayor de 3K, el poke a 1026 sería: POKE 1026,18. Si el VIC estaba ampliado sólo con los 3 K de RAM, el programa cargará perfectamente sin necesidad de teclear la línea cero y los pokes.

Los programas del VIC se pueden utilizar en los PETs 3000 mientras no empleen instrucciones POKE a la memoria de pantalla, y prescindiendo de los pokes de sonido. La pantalla en el PET 3000 se encuentra a partir de la dirección 32768 ($8000), mientras que en el VIC 20 no ampliado o sólo con la ampliación de tres K RAM está situada en la dirección 7680 ($1E00), y si está ampliado con más memoria se coloca en las direcciones 4096($1000) y siguientes. Si deseáis, sabiendo la diferencia entre las posiciones que ocupan la memoria de pantalla en uno y otro ordenador, podéis utilizar en los PETs los programas que publicamos para el VIC con las modificaciones que necesite.

Para aquellos que dispongan de un CBM 700, les puede ser útil el programa ARRIBA PERISCOPIO publicado en el número 2. Este programa era un desensamblador y un pequeño monitor de lenguaje máquina, y funciona perfectamente en el 700 con sólo añadir una línea al principio del programa en la que especifiquemos el banco de memoria en el que vamos a trabajar, puede ser: 500 BANK 15, o hacer un INPUT para escogerlo y luego la instrucción BANK.

Como información adicional para los propietarios del 700, que todavía no lo sepan, la CPU que utiliza es la 6509, que como el 6502, y 6510 puede direccionar solamente 64 K de memoria, pero el 6509 incorpora dos registros en las direcciones 0 y 1 que le permite utilizar la técnica de bancos de direccionamiento, con la que puede acceder a 16 de ellos, cada uno con 64 K, lo que hace un total de 1.048.576 Bytes direccionables (1 Mbyte).

Los bancos se numeran del 0 al 15, y están asignados como sigue:

Banco 0 reservado para el futuro ???

Banco 1 reservado para el texto de los programas en Basic.

Banco 2 almacena los datos (variables, arrays, strings).

Banco 3 al 14 sirven para ampliaciones de RAM.

Banco 15 reservado para el sistema (Kernal, DOS, Basic, etc.).

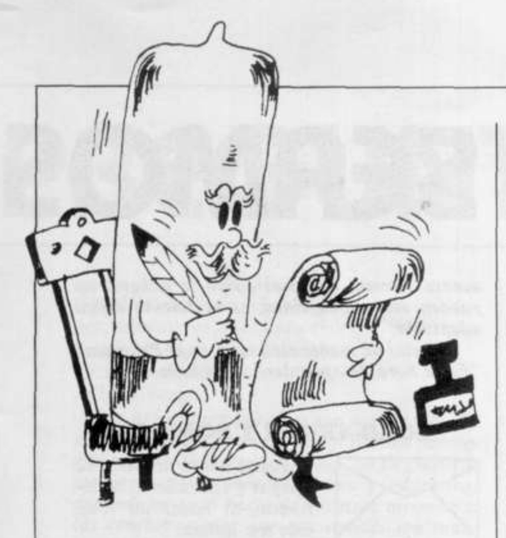
Puede parecer poco el disponer de sólo 64 K de RAM para programas en Basic, pero si tenéis en cuenta que son sólo para el programa, y que las variables se almacenan en otros bancos, podréis ver que es mucha memoria. Como dato curioso, diré que el programa de control de suscriptores que utilizamos en la revista, está escrito en Basic, utiliza ficheros indexados relativos, permite una infinidad de opciones (caducidad, modo de pago, número de cheque, reembolso o talón, etc.) y ocupa un poco menos de 10 K de RAM. Podría ocupar menos de 4 K RAM, pero le añadí montones de sentencias REM por todas partes para que si alguien quiere modificarlo en el futuro pueda hacerlo sin dificultad.

El banco 15 del 700 es el más interesado, pues en él encontramos todos los dispositivos de entradas/salidas, kernal, etc. Un dato curioso también es que el 700 utiliza el mismo chip generador de sonido que el Commodore 64, por lo que podríamos utilizarlo para aplicaciones musicales como su hermano pequeño, para ello debemos saber dónde se encuentra el SID (MOS 6581) y hacer los pokes correspondientes como los hacemos en el C64. Para que sepáis dónde se encuentra "escondido" el chip, publicamos una lista de direcciones del banco número 15 del 700.

DIRECCION	CONTENIDO
$0000	***Registro de ejecución del 6509.***
$0001	***Registro de indirección del 6509.***
$0002-$03FF	***RAM del sistema, punteros kernal, etc.***
$0400-$07FF	***RAM libre***
$0800-$0FFF	***RAM utilizada por el sistema***
$1000-$1FFF	***Reservado para los discos integrados***
$2000-$7FFF	***24 K RAM o ROM de cartucho***
$8000-$BFFF	***ROM intérprete basic 4.75***
$D000-$D7FF	***RAM de pantalla 80×25.***
$D800-$D8FF	***Controlador de vídeo Motorola 6845.***
$D900-$d9FF	***MOS 6523 TPI para discos integrados.***
$DA00-$DAFF	***MOS 6581 SIS chips de sonido***
$DB00-$DBFF	***MOS 6526 CIA interface para Z80 o 8088***
$DC00-$DCFF	***MOS 6526 CIA***
$DD00-$DDFF	***MOS 6551 ACIA para interface RS-232***
$DE00-$DEFF	***MOS 6525 TPI#1.***
$DF00-$DFFF	***MOS 6525 TPI #2***
$E000-$EFFF	***ROM editor de pantalla.***
$F000-$FFFF	***ROM kernal***

Si queréis pasar los programas musicales del 64 al 700 debéis cambiar las direcciones $D400-$D7FF al C64 a las direcciones $DA00-$DAFF del 700 en el banco 15.

Con estos detalles damos por terminada esta serie y dejamos paso a algunas series más interesantes que nos habéis pedido, la primera de ellas comenzará en el próximo número y es una serie de tres capítulos para un gran sector de nuestro público... los radioaficionados, programas para utilizar el C-64 y VIC-20 en su interesantísimo campo.



CARTA BLANCA

FALLOS DE CALCULO



El programa que mando redondea los resultados con los decimales que queramos. Sin embargo, sucede a veces (muy pocas veces afortunadamente), que los resultados aparecen con todos los decimales. He pasado bastantes ratos intentando entender por qué sucede esto y no he encontrado respuesta, por lo que me gustaría que resolviéseis este problema. El método que utilizo para redondear los decimales es el siguiente:

Si deseamos el resultado con 3 decimales y A1 representa un número con 8 decimales, el mismo número con 3 decimales será (A2)

MM = (10) ↑ 3

A2 = INT (A1MM*+.5)/MM

Rafael Muñoz Alfonso
C/Costa Brava, 15 (Can Batlle)
Vallirana (Barcelona)

El método que nos envías es el correcto.

El fallo está en la forma en la que el ordenador realiza las operaciones aritméticas internas, redondeando a veces.

Por ejemplo:

PRINT COS (2*π) nos dará 1

Sin embargo, el INT(COS(2*π)) ... ¡da cero!

El ordenador ha hallado el coseno de 2 Pi. Para él, es 0,9999999999... lo que hace es redondearlo a 1 cuando lo va a imprimir en pantalla. Naturalmente, el INT de 0.99999... es cero. Este tipo de errores es bastante difícil de evitar. Por ejemplo, podíamos hacer.

En vez de: IF INT(COS(2 * π))=1 then...

Hacemos:

IF VAL (STR$(INT(COS(2*π))))= 1 then...

MUSICA MAESTRO



Me gustaría me dijeran dónde podría adquirir el libro del Guía del Programador y qué precio tiene.

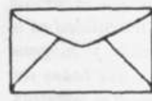
En el programa publicado en el número 6 MUSICA MAESTRO me gustaría saber si es imprescindible tener impresora o no, de lo contrario ese programa a mí no me funciona.

Josep M.ª Fondevila
Suscriptor 5130

La Guía del Programador vale unas 3.000 ptas. Puede comprarse en la mayoría de las tiendas especializadas en informática que existen en las ciudades españolas. De todos modos, consulta con Microelectrónica y Control pues ellos podrán darte una mayor información.

Una de las opciones del Programa Musica Maestro era sacar por la impresora los datos de lo que en ese momento estaba en la pantalla. No es imprescindible tener impresora. Si no te funciona será debido a algún fallo en otra zona del Programa.

LENGUAJES PARA EL VIC



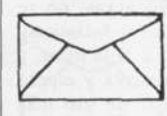
Quisiera saber si se disponen de otros lugares de programación para el VIC y de ser así me gustaría saber los precios del mercado.

También me gustaría saber si se ha publicado en España el libro "6502 Programing manual" y en qué editorial.

Ricardo Montañana Gómez, 17 años
C/Horno nº 4, 1º
Madriguera (Albacete)

Un lenguaje disponibe para el VIC-20 es el FORTH, pero no tenemos por el momento la información sobre precios, prestaciones, etc. El libro que citas lo hemos visto publicado en España pero en INGLES.

MAS EDUCATIVOS Y MENOS MARCIANITOS



Quiero deciros que me gusta vuestra revista, pero que sería conveniente aumentar los programas educativos, pues ya pasamos de marcianitos y cosas por el estilo.

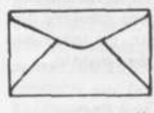
¿Me podéis decir de algún programa sobre música para el VIC sin ampliación?

David Moreno
C/Lleida, 6, 3
Sant Joan Despi
(Barcelona)

Ya hemos contestado a una petición similar a la tuya en este mismo número, el tipo de programas que aparecen en la revista depende en gran parte de lo que los mismos colaboradores nos envíen, de modo que ya sabéis: MANDARNOS MAS PROGRAMAS EDUCATIVOS.

Conocemos algún programa musical con el VIC SYNTHESIZER, pero es mejor que pidas información directamente a las casas que distribuyen soft, ellos te enviarán mucha más información.

¿VIC + 64 KRAM = C-64?



Soy suscriptor vuestro desde hace 1 año más o menos, y aprovecho esta ocasión para preguntaros 2 cuestiones que quizás parezcan muy "chorras", pero siempre tengo esta duda y quiero aclararla de una vez por todas.

La primera es: Lleváis ya 2 ó 3 meses anunciando para el VIC-20 tarjetas de ampliación de memoria. Una es de 64K RAM + 2 K EPROM (En la página 56 del número 9 lo encontraréis). Desearía saber si con esta tarjeta conseguiré tener un VIC-20 transformado en un C-64, si es así, si los cassettes para el C-64 irían bien para mi VIC.

La segunda es cuando nombráis el Superexpander ¿qué es?, un cartucho de ampliación de memoria determinado, o es simplemente el cartucho de 3,5K RAM, de 16K RAM, etc.

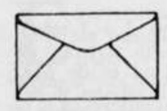
Carlos Darder Tobat
C/Magallanes C'anlitos, s/n
(Son Vida)
Palma de Mallorca

El VIC-20 sigue siendo un VIC-20 por mucha memoria que le añadas lo que pasa es que amplías su capacidad de almacenamiento, y con ello el tamaño de los datos y programas que puedes manejar con el, pero nunca se convierte en un C-64 a menos que cambies casi todos los circuitos que lleva dentro (la CPU, el controlador de vídeo, las RAMs, las ROMs, las CIAs, etc.) Por tanto, en general, no

podrás cargar los programas de un C-64, en un VIC-20.

Superexpander es el nombre que la casa Commodore le dio a un cartucho de ROM que añade comandos gráficos y de sonido al VIC-20, en otros países lo puedes encontrar con una ampliación de memoria de 3K RAM incorporada o sin ella, pero en España normalmente sólo se vende con ella, por tanto siempre que mencionemos el Superexpander, ya sabes que es ese cartucho que tiene nombre propio.

ERRORES EN EVALUACION DE COLEGIOS

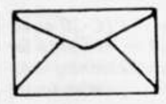


Les agradecería que aclararan el por qué el Programa Colegios que venía en el número 5 no funciona. De acuerdo con el manual del floppy disk Vic-1541, las sentencias 2006, 2007, 2090, 2095, 2710, 2715, 2845, 2860, 2945, 2960, 5120, 5130, 5220, 5230 y 5240 están mal y el programa así lo acusa al ponerlo en marcha. Soy profesor de Enseñanza Media y tengo gran interés en este Programa (el cual, dicho sea de paso, es muy difícil de seguir en su estructura por la gran cantidad de GOTO y GOSUB que tiene).

Fco. Herrera Arrocha
C/Ruiz de Alda, 36, 2º. pta. 4
35007 Las Palmas

La sintaxix de estas líneas es correcta, aunque no coincide con la que pone el manual en español, no da errores, y el programa funciona perfectamente. Si consigues un manual americano y otro alemán, compáralos con el manual inglés y el español que vienen con la unidad de discos, de este modo podrás comprobar que ninguno de ellos coincide en la sintaxis de muchos comandos, esto es en parte por errores de imprenta, y en otra parte porque la unidad de discos 1541 tiene una gran flexibilidad en la forma de admitir los comandos y ninguna manera es mejor que la otra, ya que todas son equivalentes (al menos que sepamos nosotros, ya que hemos probado todas las formas de los comandos y no apreciamos diferencias).

CONTABILIDAD EN CINTA



El programa "LLEVA TU PROPIA CONTABILIDAD", ¿se puede adaptar a la cassette, y cómo?

Los datos grabados en una cassette puedo sacarlos en pantalla ¿PERO COMO PUEDO OPERAR CON ELLOS? Gracias.

Haciéndome eco de tus palabras, no se podrían hacer más programas didácticos, formulación de química, teoría de la física, y un largo, etc. Gracias nuevamente y espero que entre todos podamos dar un uso más interesante a los Microordenadores, que aparte de entretenernos nos sirvan para aprender un poco más.

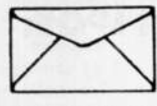
José Manuel Solanas Masip
C/Labadoras, 6, 3º Dcha.
Teléf. 229 630
26005 Logroño

El programa "Contabilidad" está hecho para que abra ficheros en DISCO. Hemos tenido problemas con gente que nos enviaba pedidos de CINTAS de este programa. Como son muchas las solicitudes, estamos trabajando en una versión para cinta. De modo que cuando hagáis vuestros pedidos, indicadnos si queréis la versión que funciona con ficheros en disco o la que funciona con ficheros en cinta.

Por supuesto publicaremos las correspondientes modificaciones para los que prefieran teclear el programa por sí mismos.

Supongo que te refieres a datos grabados como ficheros. El manual explica claramente cómo abrir ficheros para grabar y leer datos.

EVALUACION DE COLEGIOS



Soy un asiduo comprador de su revista y debo felicitarle por la calidad de la misma. Tengo un Commodore 64 comprado recientemente y por tanto soy nuevo en la micromanía por ello necesito toda la ayuda de quien pueda dármela, me animo a escribirle para plantearle tres preguntas:

—En la revista nº 5 editan un programa de evaluación de colegio ¿se puede adaptar a cinta? ¿qué hay que variar?

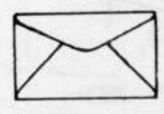
—¿Cuándo editarán un programa para un reloj de manecilla, he visto algunos en otras revistas pero para otros ordenadores y no sé adaptarlos, espero su contestación, y sobre todo háganla sencillita, para que pueda comprenderla.

Cipriano López Viegas
Edificio San Jorge, nº 10-2º 4ª
Badalona (Barcelona).

El programa "Evaluación de Colegios" funciona sólo con disco. Porque usa ficheros de acceso directo. Como este tipo de ficheros no pueden abrirse en cinta, es realmente difícil adaptarlo.

Un reloj de manecillas salió en el Programa "Es la hora de aprender" del número 7.

LOAD-ERROR Y SONIDO



Algunas veces después de apagar y encender el ordenador, al hacer un load este me indica.

FOUND "XXXX"
?LOAD ERROR
OUT OF MEMORY

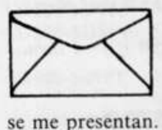
Pero el ordenador todavía no ha empezado a cargar el listado del programa. ¿Cómo puedo solucionarlo?

2) Cuando saco por pantalla el listado de un programa, o mensajes al correr un programa, o realizo dibujos con el Super Expander, el ordenador emite un pitido que va aumentando si coloco más caracteres en la pantalla, sobre todo si los caracteres son en negro, y va disminuyendo si los quito. El Poke de sonido está en 0, y lo hace con varios televisores. ¿Cómo puedo quitarlo? ¿Tiene el VIC-20 algún mando para regularlo como el C-64? (mi ordenador es un VIC-20).

Xavier Naval Marcos
Paseig Pere III, 80 2.ª
Manresa
Barcelona

1. Esto se debe a algún fallo en la cinta o en el azimut del datassette. En el artículo "Qué hacer con tu datassette para que entren esas cintas rebeldes" que salió publicado en el número 8, puede que encuentres la forma de arreglarlo. 2. A veces suele suceder que no son del todo compatibles el ordenador y la televisión, o que el modulador del ordenador no está bien ajustado. Hay una bobina dentro del modulador que permite solucionarlo, pero antes de intentarlo es mejor que consultes con Microelectrónica y Control.

BALONCESTO



Su revista es de mi agrado; llevo poco tiempo metido en el mundo de los ordenadores y esta revista me soluciona problemas que se me presentan.

Querría me contestasen si en España se vende un cartucho de baloncesto, pues sé que en el extranjero ya está a la venta, pero no he oido nada sobre si había llegado a España.

Jesús Pérez

No nos dices qué ordenador tienes, pero sabemos que existe al menos un programa en disco para el C-64 llamado BASKETBALL. Esperamos que el distribuidor nos lo mande para que podamos daros una mayor información a todos.

CONSULTA SOBRE UNA RESEÑA



Primero quiero felicitaros por la revista tan buena que hacéis. Quiero preguntaros si los programas de juegos que anunciábais en el número 4 de la revista para el VIC-20, que son: El camarero Loco, El superdetective y Hechizado, necesitan periféricos, como pueden ser: Joystick, Ampliaciones de

PREGUNTONES ¿ ?

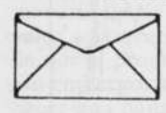


memoria o Paddes. También quiero deciros si podéis publicar más programas de utilidades, educativos, etc. para el VIC-20.

Julián M. Manglano García
C/Acera de Canasteros, 16 4ª
18012 Granada

Los programas que citas son para el VIC-20 sin ampliación de memoria, y pueden funcionar con joystick o con teclado. Lo de publicar menos juegos y más utilidades y educativos depende en gran parte de vosotros mismos; si lás colaboraciones que nos enviáis son todas juegos, la mayoría de los programas publicados serán juegos, si nos enviáis más utilidades, las publicaremos. Todo esto es así por ser esta revista, una revista hecha por los propios usuarios y colaboradores.

TECLADO MUSICAL



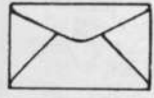
¿Qué instrucción/es es/son necesarias para que al apretar una tecla cualquiera del C64 se produzca un sonido.

Josep Maria Lecuona i Aragones
c/Sant Miguel, 9
Sant Feliu de Gixols
Gerona

Para conseguir esto, debes tener una rutina en código máquina o en basic en el ordenador que active las notas al pulsar las teclas. En el manual de usuario del C-64 aparece publicada una que se llama piano (o algo parecido)

No existe ningún comando que te permita hacerlo en modo directo.

INSERT O DELETE



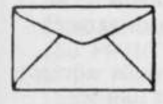
Señores: Después de felicitarles por el acierto de nuestra revista paso a hacerles una pregunta.

Con mucha frecuencia al pulsar la tecla INS-DEL/esta queda bloqueada sin moverse el cursor hacia la izquierda. Es normal ¿qué hacer en este caso?

Jordi Erra i Fabregat
Pau Viè, 52
Sant Cugat del Valles
Barcelona

La verdad es que no es normal, en todo caso sería así si estuvieras pulsando la tecla SHIFT simultáneamente.

RELOJ EN PANTALLA



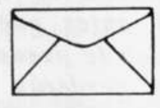
Desearía si fuera posible me dijeran cómo poner en un programa (por ejemplo de juegos) un reloj, mejor dicho, que en un rincón del T.V. mientras estuviera el juego en marcha fuera marcando la hora, minutos y segundos.

Para eso, entre otras cosas, sirve alguna rutina similar a la rutina en CM que salió publicada en el número 9 en "Cursillo de Lenguaje Máquina". Si quieres hacer lo mismo en basic, puedes:

1) En un lugar por el que pase siempre el programa: PRINT "[HOME]"; TI$

2) Hacer una rutina como: 2000 PRINT [HOME]" TI$ y llamarla de vez en cuando. Quizás Diego lo saque en el cursillo de lenguaje máquina próximamente.

COLABORACIONES Y ENVIOS



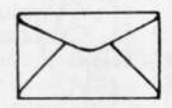
En primer lugar les felicito por esta revista, y después que tengo hechos dos juegos, y exactamente no sé cómo mandarlos.

Benito Santed Pérez
Pº Pujadas, 17-19-7º 2.ª
Barcelona 08018

Los que deseéis: mandarnos programas de colaboraciones, podéis hacerlo en cinta o en disco, este es el mejor método, pero si prefieres también podéis mandar los listados por impresora, aunque en este caso se retrasará su publicación algo más.

Las cintas que envíeis se devolverán con algún programa de regalo (de los de la revista, no programas comerciales).

TINY FORTH



El lenguaje Tiny Forth ¿qué es? ¿Es una versión barata del Forth, o es algo totalmente diferente?. Así, es compilado o interpretado.

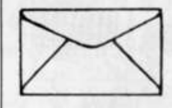
Ivo Pla Vallvé
c/Capuchinos, 30, 4
Tarragona

Sabemos de dos versiones Inglesas del Tiny Forth, una de ellas es una implementación en Basic del lenguaje forth, y permite hacer tus primeros pinitos en este lenguaje desde tu ordenador con sólo teclear un pequeño programa que fue publicado en la revista Computing Today, con esto puedes conocer el Forth antes de gastar un montón de dinero en un cartucho que te permita trabajar este lenguaje de verdad.

La otra versión de Tiny Forth, es también inglesa, y es en realidad un cartucho para trabajar en Forth, son solamente 4K ROM, por lo que su vocabulario es muy limitado (normalmente son 16K ROM), pero con el Forth es un lenguaje que te permite definir tu propio vocabulario, es suficiente con disponer de unas pocas primitivas para conseguir crear tu Forth particular.

Como imagino que sabrás, el Forth no es siempre interpretado ni siempre compilado, depende de cómo tú ordenes que se definan las palabras forth. Naturalmente el Tiny forth que está implementado en basic, es una simulación de forth lenta y que trabaja bajo el control del intérprete Basic.

CATALOGOS



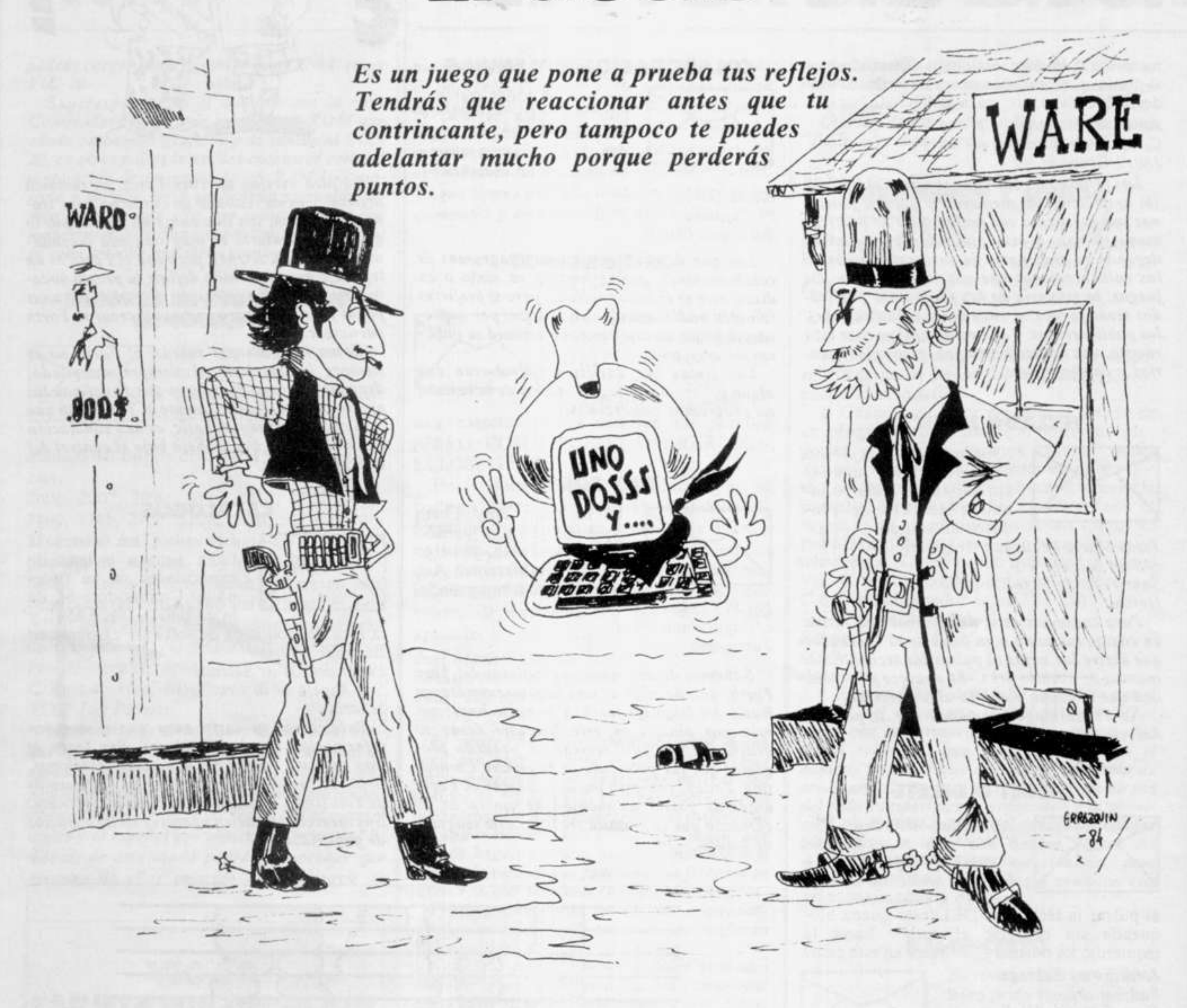
Les ruego me envíen la ficha técnica y lista de capacidades de su ordenador Commodore-64, de la unidad de disco 1541, y la lista de biblioteca de software existente en España.

Pablo del Casar Serrano
Sor Angela de la Cruz, 10
Madrid

Publicamos tu carta para que te manden información los distribuidores. Por favor, y esto va para todos: No nos pidáis catálogos, listas de juegos, etc. Nosotros no disponemos de esas listas. Hacedlo a las casas de software. Sus direcciones suelen aparecer en los espacios de publicidad.

El duelo

Es un juego que pone a prueba tus reflejos. Tendrás que reaccionar antes que tu contrincante, pero tampoco te puedes adelantar mucho porque perderás puntos.



Por David D. BUSCH (RUN. EE.UU.)

Traducido por Valerie SHANKS y adaptado por Alvaro IBAÑEZ

Te apetece poner tus reflejos a prueba contra los del ordenador? No sería muy justo —el C 64 y el VIC-20 son capaces de contar hasta 50 antes de que tú puedas tocar ni una tecla. Los ordenadores son menos inteligentes y, desde luego tienen menos capacidad de movimiento que los seres humanos, pero si les das una tarea sencilla que realizar, lo harán más rápido que cualquiera.

Entonces, con este juego puedes poner tus reflejos a prueba contra los de otro ser humano, es decir realizar un duelo y el ordenador actuará de padrino. Esto parece más razonable, ya que el ordenador no tiene preferencias y podrá calcular la velocidad de respuesta más rápido que tú. Te puedes aprovechar de la velocidad del ordenador sin que ella se aproveche de ti.

En este juego, tú y tu contrincante tenéis que esperar una señal, que indicará que los dos tenéis que intentar a pulsar la tecla que le ha sido asignada. El jugador de la derecha tiene que pulsar la tecla INST/DEL. El que llegue primero se lleva un punto.

Una vez dada la señal, si tú pulsas cualquier otra tecla no significa ni que pierdas tu turno ni que tengas posibilidades de ganar. Sin embargo, si pulsas cualquier tecla antes de que se dé la señal ¡¡AHORA!! pondrás en marcha una rutina que resta un punto de tu puntuación. El que llegue primero a diez puntos gana el juego.

Los dos jugadores son identificados como el de la izquierda y el de la derecha, y las puntuaciones de cada uno se almacenan en las variables PL y PR, respectivamente. En cada partida, el ordenador elige un tiempo de retraso aleatorio.

En vez de medir los "jiffies", intervalos de 1/60 de segundo, el ordenador cuenta desde uno hasta un número dado, siendo este intervalo de tiempo nuestro tiempo de retraso. Para este juego, el ordenador cuenta al azar una cantidad de números —más de 500, pero menos de 1.500.

C-64 y VIC-20

Entonces se inicia el bucle For...Next que afecta el tiempo de retraso. Cada vez que se pasa por este bucle, el ordenador comprueba para ver si alguien se adelante y pulsa una tecla.

Aunque los ordenadores Commodore pueden hacer uso de varios métodos para realizar esto, uno que no se utiliza es hacer un Peek en una posición de memoria determinada. Lo que hace este juego es aprovecharse de un truco muy popular de programación que resulta bastante más sencillo para que lo pueda comprender el principiante.
Esta técnica es una línea bastante rara de Get A$. A diferencia de la mayoría de las líneas Get, esta no se repite hasta que el ordenador pulsa una tecla. La forma más común es la siguiente:
10 GET A$: IF A$=" " GOTO 10
Cuando encuentra una línea así, el ordenador busca un solo carácter de teclado y asigna su valor o la variable A$. Si no se pulsa ninguna tecla, A$ será igual a nada, o un "string" nulo. Esto también se expresa como dos comillas (" ") que no encierran nada.
Normalmente, el ordenador pasará a la siguiente sentencia del programa. Dado que el C-64 y el VIC-20 repasan el teclado tan de prisa, sería muy difícil llegar a pulsar una tecla justo en el momento en que el ordenador pide una entrada de datos. Por lo tanto, las sentencias Get normalmente se colocan dentro de los bucles, como el que hemos visto antes.
En caso de que A$ no tiene ningún valor (" "), que sería la mayoría de las veces, se repetirá la línea 10. A$ tendrá un valor solamente cuando se pulsa una tecla, con el resultado de que el programa pasa a la siguiente línea.
Sin embargo, en el juego "El Duelo", el ordenador no debe hacer una pausa esperando que pulses una tecla. Tiene que seguir contando el tiempo de retraso elegido, y vigilar a la vez que no se pulse ninguna tecla. En vez de dejar que el ordenador se quede colgado en una línea, podrías escribir un módulo como el siguiente.

100 FOR N= TO DELAY
110 GET A$: IF A$ < > " " GOTO 140
120 NEXT N
130 END
140 PRINT "DISPARO DEMASIADO PRONTO"

Con este módulo, el ordenador pasará por el bucle For...Next a una velocidad normal. Si pulsas una tecla al ejecutarse la línea 110, el programa se bifurca a la línea 140, y aparece un mensaje. De no ser así, dado que A$ es igual a " ", el programa pasará a la línea 120 para repetir el bucle. Ya que cada operación se realiza tan de prisa, parece que el ordenador está repasando el teclado constantemente, aunque está llevando la cuenta del bucle de retraso a la vez.
Esta es la técnica que se utiliza en "El Duelo". Se incluye la línea Get A$, y si A$=" ", el programa pasa directamente a Next N.
Sin embargo, si AS < > " ", el programa pasa a la línea 550, comprueba para ver quién era el culpable y resta un punto de su puntuación. Para que veas, durante el bucle de retraso no se puede pulsar una tecla.
Si el bucle llega a su fin sin ninguna interrupción, la palabra ¡¡AHORA!! aparece en pantalla. Se inicia una señal acústica dentro de un bucle más largo Get A$. Esto se repite hasta que se pulsa una tecla. El código ASCII de dicha tecla se almacena en la variable A.
Si la tecla de flecha a la izquierda o la tecla INST/DEL ha sido pulsada el programa pasa a la rutina de ganador. De no ser así, la tecla se ignora y el programa espera una entrada aceptable.
A medida que tú y tu contrincante van ganando, se van acumulando puntos, y una flecha le apunta al ganador y se sigue jugando. Al final del juego, se anuncia el ganador.

VIC-20 SIN EXPANSION

```
10 REM ************
20 REM *          *
30 REM * EL DUELO *
40 REM *          *
50 REM ************
60 MA=22
70 POKE36878,15
80 VOICE=36874
90 SO=255
100 PRINT"[CLR]"
110 PRINTTAB(6)"[RVSON][RED]EL[SPC]D
UELO[BLU][2CRSRD]"
120 PRINT"[2SPC]CUANDO[SPC][RVSON]AH
ORA![RVSOFF][2SPC]"
130 PRINT"[CRSRD][SPC]APAREZCA,EL[SPC]
JUGADOR"
135 PRINT"[CRSRD][SPC]DE[SPC]LA[SPC]
IQUIERDA[SPC]DEBE"
140 PRINT"[CRSRD][SPC]PULSAR[SPC][RVSON]
←[RVSOFF],[SPC]Y[SPC]EL[SPC]DE"
145 PRINT"[CRSRD][SPC]LA[SPC]DERECHA
[SPC][RVSON]INST[RVSOFF]"
150 PRINT"[2CRSRD][SPC]EL[SPC]PRIMER
O[SPC]QUE"
155 PRINT"[CRSRD][SPC]LLEGUE[SPC]A[SPC]
10[SPC]PUNTOS"
157 PRINT"[CRSRD][SPC]SERA[SPC]EL[SPC]
GANADOR!!"
160 PRINTTAB(4)"[CRSRD][RVSON][CYN]P
ULSA[SPC]UNA[SPC]TECLA[BLU]"
170 GETA$:IFA$=""THEN170
180 PRINT"[CLR][2CRSRD]";TAB(2)"PULS
A[SPC]RETURN[SPC]PARA"
190 PRINTTAB(2)"SEGUIR."
200 INPUTA$
210 PRINT"[CLR]"
220 IFPL>9ORPR>9THEN690
230 DELAY=RND(1)*1000+500
240 FORN=1TODELAY
250 GETA$:IFA$<>""THEN550
260 NEXTN
270 PRINT"[3CRSRD]";TAB(MA/2-4);"[RVSON]
[BLU]AHORA![BLU]"
280 GETA$
290 POKEVOICE,SO
300 POKEVOICE,0
310 IFA$=""THEN280
320 A=ASC(A$)
330 IFA=95ORA=20THEN350
340 GOTO280
350 IFA=95THEN450
360 REM
370 PRINT"[CLR][2CRSRD]"
380 PR=PR+1
390 PRINTTAB(4)"IZQUIERDA:";PL
400 PRINTTAB(4)"DERECHA:";PR
410 PRINTTAB(4)"[2CRSRD]GANADOR[SPC]
[RVSON][RED]--->[RVSOFF][BLU]"
420 A$=""
430 FORN=1TO1000:NEXT
440 GOTO180
450 PRINT"[CLR][2CRSRD]"
460 PL=PL+1
470 PRINTTAB(4)"IZQUIERDA:";PL
480 PRINTTAB(4)"DERECHA:";PR
490 PRINT"[2CRSRD][RVSON][RED]";TAB(
4)"<---[RVSOFF][BLU][SPC]GANADOR"
500 PRINT"[2CRSRD]"
510 IFA$<>""THEN420
520 IFA$<>""THEN520
530 FORN=1TO1000:NEXT
540 GOTO180
550 A=ASC(A$)
560 IFA=95ORA=20GOTO580
570 GOTO260
580 IFA=95THEN600
```

```
590 GOTO630
600 PRINT"[CRSRD][SPC]EL[SPC]JUGADOR
[SPC][RVSON][RED]<---[RVSOFF][BLU]":
PRINT"[CRSRD][SPC]DISPARO[SPC]DEMASI
ADO"
610 PRINT"[CRSRD][SPC]PRONTO!":PRINT
"[CRSRD][SPC]PIERDE[SPC]UN[SPC]PUNTO
.":PL=PL-1
620 GOTO660
630 PR=PR-1
640 PRINT"[CRSRD][SPC]EL[SPC]JUGADOR
[SPC][RVSON][RED]--->[RVSOFF][BLU]":
PRINT"[CRSRD][SPC]DISPARO[SPC]DEMASI
ADO"
650 PRINT"[CRSRD][SPC]PRONTO!":PRINT
"[CRSRD][SPC]PIERDE[SPC]UN[SPC]PUNTO
.":PR=PR-1
660 PRINTTAB(6)"[2CRSRD][RVSON][CYN]
PULSA[SPC]UNA[SPC]TECLA[BLU]"
670 GETA$:IFA$=""THEN670
680 GOTO180
690 PRINT"[CLR][2CRSRD]"
700 IFPL>9THENPRINT"EL[SPC]JUGADOR[SPC]
[RVSON][RED]<---[RVSOFF][BLU][SPC]GA
NA!":GOTO720
710 PRINT"EL[SPC]JUGADOR[SPC][RVSON]
[RED]--->[RVSOFF][BLU][SPC]GANA!"
720 PRINT"[2CRSRD]"
730 PRINT"OTRA[SPC]PARTIDA?"
740 INPUTB$
750 IFLEFT$(B$,1)="S"THENRUN
```

C-64

```
10 REM ************
20 REM *          *
30 REM * EL DUELO *
40 REM *          *
50 REM ************
60 MA=40
70 POKE53281,1
100 PRINT"[CLR]"
110 PRINTTAB(17)"[RVSON][RED]EL[SPC]
DUELO[BLU][2CRSRD]"
120 PRINTTAB(8)"[2SPC]CUANDO[SPC][RVSON]
AHORA![RVSOFF][2SPC]"
130 PRINTTAB(8)"[CRSRD][SPC]APAREZCA
,EL[SPC]JUGADOR"
135 PRINTTAB(8)"[CRSRD][SPC]DE[SPC]L
A[SPC]IQUIERDA[SPC]DEBE"
140 PRINTTAB(8)"[CRSRD][SPC]PULSAR[SPC]
[RVSON]←[RVSOFF],[SPC]Y[SPC]EL[SPC]D
E"
145 PRINTTAB(8)"[CRSRD][SPC]LA[SPC]D
ERECHA[SPC][RVSON]INST[RVSOFF]"
150 PRINTTAB(8)"[2CRSRD][SPC]EL[SPC]
PRIMERO[SPC]QUE"
155 PRINTTAB(8)"[CRSRD][SPC]LLEGUE[SPC]
A[SPC]10[SPC]PUNTOS"
157 PRINTTAB(8)"[CRSRD][SPC]SERA[SPC]
EL[SPC]GANADOR!!"
160 PRINTTAB(15)"[CRSRD][RVSON][CYN]
PULSA[SPC]UNA[SPC]TECLA[BLU]"
170 GETA$:IFA$=""THEN170
180 PRINT"[CLR][2CRSRD]";TAB(2)"PULS
A[SPC]RETURN[SPC]PARA"
190 PRINTTAB(2)"SEGUIR."
200 INPUTA$
210 PRINT"[CLR]"
220 IFPL>9ORPR>9THEN690
230 DELAY=RND(1)*1000+500
240 FORN=1TODELAY
250 GETA$:IFA$<>""THEN550
260 NEXTN
270 PRINT"[3CRSRD]";TAB(MA/2-4);"[RVSON]
[BLU]AHORA![BLU]"
280 POKE54296,15
290 POKE54273,1
300 GETA$
310 POKE54296,0
320 IFA$=""THEN280
330 A=ASC(A$):IFA=95ORA=20THEN350
340 GOTO280
350 IFA=95THEN450
360 REM
370 PRINT"[CLR][2CRSRD]"
380 PR=PR+1
390 PRINTTAB(4)"IZQUIERDA:";PL
400 PRINTTAB(4)"DERECHA:";PR
410 PRINTTAB(4)"[2CRSRD]GANADOR[SPC]
[RVSON][RED]--->[RVSOFF][BLU]"
420 A$=""
430 FORN=1TO1000:NEXT
440 GOTO180
450 PRINT"[CLR][2CRSRD]"
460 PL=PL+1
470 PRINTTAB(4)"IZQUIERDA:";PL
480 PRINTTAB(4)"DERECHA:";PR
490 PRINT"[2CRSRD][RVSON][RED]";TAB(
4)"<---[RVSOFF][BLU][SPC]GANADOR"
500 PRINT"[2CRSRD]"
510 IFA$<>""THEN420
520 IFA$<>""THEN520
530 FORN=1TO1000:NEXT
540 GOTO180
550 A=ASC(A$)
560 IFA=95ORA=20GOTO580
570 GOTO260
580 IFA=95THEN600
590 GOTO630
600 PRINT"[CRSRD][SPC]EL[SPC]JUGADOR
[SPC][RVSON][RED]<---[RVSOFF][BLU]":
PRINT"[CRSRD][SPC]DISPARO[SPC]DEMASI
ADO"
610 PRINT"[CRSRD][SPC]PRONTO!":PRINT
"[CRSRD][SPC]PIERDE[SPC]UN[SPC]PUNTO
.":PL=PL-1
620 GOTO660
630 PR=PR-1
640 PRINT"[CRSRD][SPC]EL[SPC]JUGADOR
[SPC][RVSON][RED]--->[RVSOFF][BLU]":
PRINT"[CRSRD][SPC]DISPARO[SPC]DEMASI
ADO"
650 PRINT"[CRSRD][SPC]PRONTO!":PRINT
"[CRSRD][SPC]PIERDE[SPC]UN[SPC]PUNTO
.":PR=PR-1
660 PRINTTAB(6)"[2CRSRD][RVSON][CYN]
PULSA[SPC]UNA[SPC]TECLA[BLU]"
670 GETA$:IFA$=""THEN670
680 GOTO180
690 PRINT"[CLR][2CRSRD]"
700 IFPL>9THENPRINT"EL[SPC]JUGADOR[SPC]
[RVSON][RED]<---[RVSOFF][BLU][SPC]GA
NA!":GOTO720
710 PRINT"EL[SPC]JUGADOR[SPC][RVSON]
[RED]--->[RVSOFF][BLU][SPC]GANA!"
720 PRINT"[2CRSRD]"
730 PRINT"OTRA[SPC]PARTIDA?"
740 INPUTB$
750 IFLEFT$(B$,1)="S"THENRUN
```

EXCURSION POR LA MICROPROGRAMACION

Cursillo de Lenguaje Máquina —IX—

Seguimos con ejemplos prácticos, y en éste número vamos a ver un ejemplo en el que le añadimos nueve comandos para manejo de gráficos al C-64. No he preparado otro programa para el VIC 20 ya que no pretendo publicar programas para los dos por igual, sólo unos "ejemplos", y esta vez le ha tocado al 64, otro día le tocará al VIC. Los que quieran hacer algo parecido en el VIC deberán disponer de alguna ampliación de RAM, no podrán copiar la ROM a RAM por no tener la misma configuración de memoria, pero podrán interceptar la rutina de error del intérprete Basic o cualquier otra rutina para añadir sus comandos, el resto de las ideas y rutinas sirven perfectamente para el VIC (relocalizándolas naturalmente).

Uno de los detalles que hasta el momento no hemos visto y que aparece en el listado del código

Por Diego ROMERO



fuente de este programa es la utilización de los MACROS. Con los macros y un ensamblador que nos permita utilizarlos (como el Macro Assembler de Commodore que distribuye Micro Electrónica y Control), podremos ahorrar tiempo al teclear un procedimiento que vamos a repetir varias veces a lo largo del programa con diferentes parámetros tecleándolo sólo una vez y luego llamando al macro.

La definición de macroinstrucción, nos dice que es una instrucción en lenguaje fuente que es reemplazada por una secuencia de instrucciones escritas en el mismo lenguaje. La macroinstrucción puede especificar valores para los parámetros que deben ser reemplazados en las instrucciones correspondientes. En los manuales de los ensambladores que permiten utilizar macros, vienen explicados más detalladamente. Con palabras sencillas, el macro es una "especie de subrutina", que permite hacer rutinas similares a ella en cualquier punto de un programa, pero la subrutina generada es distinta según los parámetros que se utilizan al llamar al macro.

A continuación paso a la traducción del artículo original de Richard Holleran, publicado en Microcomputing EE.UU. (Marzo 1984).

Saca tu C-64 de su concha

Añádele nueve comandos del Turtle Graphics al repertorio de tu Commodore 64

Por Richard HOLLERAN (Microcomputing EE.UU.)
Traducido y adaptado por Diego ROMERO

Existen dos aspectos interesantes en mi programa de gráficos: lo que es y cómo lo hace. Este programa añade nueve comandos del Turtle Graphics al Commodore 64, incluyendo Draw, Plot, Move, Left y Right. Cuando ejecutas el programa, coloca las rutinas de lenguaje máquina en la parte superior de la memoria, dejando unos 21 Kb de memoria libre.

Como probablemente sabréis, el Turtle Graphics es una configuración muy popular de gráficos por ordenador en el cual una "tortuga" (turtle) (real o imaginariamente representada) es programada para moverse por la pantalla, dejando un rastro visible tras de sí. En mi versión la tortuga es, desafortunadamente, imaginaria.

Un vistazo a los comandos

El comando DRAW inicializa la localización y orientación de la tortuga y borra la pantalla. DRAW también fija los colores azul oscuro de fondo y azul claro para el dibujo.

GR#X permite pasar de la pantalla de texto (GR#0) a la de gráficos (GR#1) y viceversa. Esto es útil si quieres volver a ver lo que tenías en la pantalla de gráficos, ya que DRAW borraría la pantalla. Si existe un error mientras estamos en el modo gráfico, el ordenador pasará al modo de texto que nos lo indicará.

COLOR X, Y permite escoger el color de la pantalla y del dibujo. Por ejemplo, COLOR 6,14 fija dibujos azul claro sobre

EXCURSION POR LA MICROPROGRAMACION

fondo azul oscuro, que son los ajustes por defecto u omisión. Los números de los colores son los mismos que aparecen en el manual del Commodore 64.

El comando MOVE mueve la tortuga imaginaria, dejando el rastro visible. Su sintaxis correcta es MOVE X, donde X es un número positivo menor que 511 o una expresión cuyo resultado sea menor que 511. Un valor mayor que 510 produce una línea de longitud imprevisible. Un valor negativo no afecta a la pantalla.

El comando LEFT X altera la orientación de la tortuga girán-

```
LINE# LOC   CODE       LINE

      0000             ; TURTLE GRAPHICS PARA EL 64
      0000             ; AUTOR: RICHARD HOLLERAN
      0000             ; ADAPTADO POR DIEGO ROMERO
      0000             ;
      0000             ;
      0000             DX     = $8000     ; CAMBIO EN DIRECCION X
      0000             DY     = $8005     ; CAMBIO EN DIRECCION Y
      0000             XCOR   = $800A     ; COORDENADA X ACTUAL
      0000             YCOR   = $800F     ; COORDENADA Y ACTUAL
      0000             CURANG = $8014     ; ORIENTACION ACTUAL
      0000             STARTX = $8019     ; VALOR X INICIAL (159)
      0000             STARTY = $801E     ; VALOR Y INICIAL (99)
      0000             STARTA = $8023     ; ANGULO INICIAL (3*PI/2)
      0000             DEGRAD = $8028     ; PI/180 PARA CONVERSION
      0000             SQUARE = $802D     ; VALOR PARA 'SQUARE'
      0000             COLOR  = $8061     ; UTILIZADO PARA 'DRAW'
      0000             FINDDX = $8085     ; PARA CALCULAR DX
      0000             FINDDY = $80A4     ; PARA CALCULAR DY
      0000             PLOT1  = $80C7     ; USADO PARA 'MOVE'
      0000             IGR0   = $8047     ; ENTRADA PARA GR#0
      0000             IGR1   = $803F     ; ENTRADA PARA GR#1
      0000             FLAG   = $97       ; LAPIZ PINTA/NO PINTA
      0000             TABLE  = $8227     ; TABLA DE PALABRAS
      0000             BITVAL = $821F     ; TABLA DE VALORES DE BITS
      0000             CHRGET = $0073     ; RUTINA GET DEL SISTEMA
      0000             IERROR = $A43B     ; RUTINA DE ERROR BASIC
      0000             READY  = $A474     ; ESCRIBE 'READY'
      0000             LET    = $A9A5     ; RUTINA 'LET' DEL BASIC
      0000             EVALEX = $AD9E     ; EVALUA EXPRESION
      0000             FIXFLO = $B391     ; CONVERSION REAL A ENTERO
      0000             INPUT2 = $B7EB     ; ENTRA DOS VALORES
      0000             PLUS   = $B867     ; SUMA EN BASIC
      0000             MULT   = $BA28     ; MULTIPLICACION BASIC
      0000             MEMFAC = $BBA2     ; MEMORIA A ACUMULADOR#1
      0000             FACMEN = $BBD7     ; FAC#1 A MEMORIA
      0000             FLOFIX = $BC9B     ; ENTERO A REAL
      0000             COSINE = $E264     ; COSENO EN BASIC
      0000             SINE   = $E26B     ; FUNCION SENO
      0000             ;
      0000                    .MAC XLOYHI  ; PREPARA REGISTROS X E Y
      0000                    LDX #<?1     ; BYTE BAJO DE DIRECCION
      0000                    LDY #>?1     ; BYTE ALTO
      0000                    .MND
      0000             ;
      0000                    .MAC ALOYHI  ; PREPARA REGISTROS A E Y
      0000                    LDA #<?1     ; BYTE BAJO DE DIRECCION
      0000                    LDY #>?1     ; BYTE ALTO
      0000                    .MND
      0000             ;
      0000                    .MAC UPDATE  ; PASA FACC#1 A MEMORIA
      0000                    XLOYHI ?1
      0000                    JSR FACMEN
      0000                    .MND
      0000             ;
      0000                    .MAC FETCH   ; PASA MEMORIA A FAC#1
```

dola X grados en contra de las agujas del reloj. RIGHT X tiene un efecto similar pero en sentido horario.

PLOT X, Y te permite activar puntos individualmente. El margen del eje x es de 0 a 319; el del eje y es de 0 a 199. El punto 0,0 está situado en la esquina inferior izquierda de la pantalla. Al igual que en el comando MOVE, los puntos fuera de la pantalla no son activados, para evitar la posibilidad de hacer un "POKE" en medio del programa que se está ejecutando. Sin embargo en el caso de PLOT, cualquier valor fuera del margen de un byte

```
LINE# LOC   CODE        LINE

00056  0000                        ALOYHI ?1
00057  0000                        JSR MEMFAC
00058  0000                        .MND
00059  0000              ;
00060  0000                        .MAC ADD            ; SUMA MEMORIA A FAC#1
00061  0000                        ALOYHI ?1
00062  0000                        JSR PLUS
00063  0000                        .MND
00064  0000              ;
00065  0000                        .MAC MULPLY         ; MULTIPLICA EL FAC#1
00066  0000                        ALOYHI ?1           ; POR LA MEMORIA
00067  0000                        JSR MULT
00068  0000                        .MND
00069  0000              ; '?1' LEE EL PRIMER PARAMETRO DE UNA
00070  0000              ; LLAMADA A UN MACRO
00071  0000              ; '#<' LEE EL BYTE BAJO, '#>' EL ALTO
00072  0000              ;
00073  0000              ;
00074  0801              *       = $0801
00075  0801  0C          ; EL PROGRAMA INCLUYE UNA LINEA BASIC
00075  0802  08          BASIC  .BYTE $0C,$08,$0A,$00,$9E,$20,$32
00075  0803  0A
00075  0804  00
00075  0805  9E
00075  0806  20
00075  0807  32
00076  0808  30                 .BYTE $30,$36,$32,$00,$00,$00
00076  0809  36
00076  080A  32
00076  080B  00
00076  080C  00
00076  080D  00
00077  080E              ; 10 SYS 2062
00078  080E  A9 37       INIT   LDA #$37         ; HABILITA ROM BASIC
00079  0810  85 01              STA $01
00080  0812  A9 A0              LDA #$A0         ; PUNTERO A ROM
00081  0814  85 FC              STA $FC
00082  0816  A0 00              LDY #$00
00083  0818  84 FB              STY $FB
00084  081A  B1 FB       LOOP1  LDA ($FB),Y      ; TRANSFIERE EL BASIC
00085  081C  91 FB              STA ($FB),Y      ; DE ROM A RAM
00086  081E  C8                 INY
00087  081F  D0 F9              BNE LOOP1
00088  0821  E6 FC              INC $FC          ; CUANDO $FC CONTIENE #$C0
00089  0823  24 FC              BIT $FC          ; QUE ES EL FINAL DE BASIC
00090  0825  50 F3              BVC LOOP1        ; ENTONCES NO SALTA.
00091  0827  A9 36              LDA #$36         ; HABILITA EL BASIC EN RAM
00092  0829  85 01              STA $01          ; BYTE ALTO DE VECTOR
00093  082B  A9 81              LDA #$81         ; DE ERROR
00094  082D  8D 01 03           STA $0301        ; VETOR DE RUTINA USR
00095  0830  8D 12 03           STA $0312        ; NUEVO ANALIZADOR.
00096  0833  8D 06 A8           STA $A806
00097  0836  A9 C4              LDA #$C4         ; BYTE BAJO DE ERROR
00098  0838  8D 00 03           STA $0300
00099  083B  A9 D2              LDA #$D2
```

EXCURSION POR LA MICROPROGRAMACION

(0-255) causa un error de "illegal quantity".

El comando TAILUP permite mover la tortuga sin dibujar una línea, probar por ejemplo: TAILUP: MOVE 50. El comando TAILDOWN baja nuevamente el "lápiz" de la tortuga.

Sería mejor utilizar los comandos originales PENUP/PENDOWN, pero pendown contiene el comando basic END, y por tanto no puede ser utilizado desde Basic.

Encuadrando

Además de estos comandos, puedes utilizar la función USR

```
LINE# LOC   CODE         LINE

00100  083D  8D 11 03          STA $0311       ; BYTE BAJO DE RUTINA USR
00101  0840  A9 DC             LDA #$DC
00102  0842  8D 05 A8          STA $A805       ; BYTE BAJO DEL ANALIZADOR
00103  0845  A9 80             LDA #$80
00104  0847  85 FE             STA $FE         ; PUNTERO DE DESTINO
00105  0849  A9 19             LDA #$19
00106  084B  85 FD             STA $FD
00107  084D  A9 08             LDA #>SOURCE    ; PUNTERO A CODIGO
00108  084F  85 FC             STA $FC         ; FUENTE.
00109  0851  A9 71             LDA #<SOURCE
00110  0853  85 FB             STA $FB
00111  0855  B1 FB       LOOP2 LDA ($FB),Y     ; MUEVE EL CODIGO
00112  0857  91 FD             STA ($FD),Y     ; AL DESTINO
00113  0859  C8                INY
00114  085A  D0 F9             BNE LOOP2
00115  085C  E6 FE             INC $FE
00116  085E  E6 FC             INC $FC
00117  0860  A5 FC             LDA $FC
00118  0862  C9 0B             CMP #$0B        ; MUEVE TRES PAGINAS
00119  0864  90 EF             BCC LOOP2       ; EN TOTAL.
00120  0866  A2 00             LDX #$00        ; CARGA X,Y CON EL NUEVO
00121  0868  A0 5C             LDY #$5C        ; FINAL DE MEMORIA LIBRE.
00122  086A  18                CLC             ; SALTA A LA RUTINA KERNAL
00123  086B  20 99 FF          JSR $FF99       ; MEMTOP Y LE DICE LA
00124  086E  4C 02 E4          JMP $E402       ; MEMORIA LIBRE AL BASIC.
00125  0871              ;
00126  0871              SOURCE
00127  0871              ;
00128  0871  88                .BYTE $88,$1F,$00,$00,$00 ; 159 EN COMA FLOT.
00128  0872  1F
00128  0873  00
00128  0874  00
00128  0875  00
00129  0876  87                .BYTE $87,$46,$00,$00,$00 ; 99 EN COMA FLOTANTE
00129  0877  46
00129  0878  00
00129  0879  00
00129  087A  00
00130  087B  83                .BYTE $83,$16,$CB,$E3,$F8 ; (3/2)*PI
00130  087C  16
00130  087D  CB
00130  087E  E3
00130  087F  F8
00131  0880  7B                .BYTE $7B,$0E,$FA,$35,$12 ; PI/180
00131  0881  0E
00131  0882  FA
00131  0883  35
00131  0884  12
00132  0885  80                .BYTE $80,$40,$00,$00,$00 ; .75
00132  0886  40
00132  0887  00
00132  0888  00
00132  0889  00
00133  088A              ;
00134  088A              GR
```

para "encuadrar" (square) un dibujo en la pantalla. Dado el formato de los puntos generados por el C-64, las líneas horizontales y verticales de igual magnitud no son de igual longitud. Prueba por ejemplo:

DRAW: FOR T=1 TO 4: MOVE 90: LEFT 90: NEXT

Dependiendo de tu monitor, esto puede parecerse más o menos a un cuadrado. En mi monitor se parece a un cuadrado cuando la línea en el eje Y es tres cuartos de la magnitud del eje X. Por tanto, el valor por defecto para USR es 0,75.

```
LINE# LOC   CODE          LINE

      088A                ;
      088A  48                    PHA               ; AVANZA PUNTERO DE TEXTO
      088B  20 73 00              JSR CHRGET
      088E  68                    PLA               ; ULTIMO CARACTER UN CERO?
      088F  C9 30                 CMP #'0
      0891  F0 0C                 BEQ GR0           ; FUE UN 1?
      0893  C9 31                 CMP #'1
      0895  D0 17                 BNE EXIT
      0897  A9 3B         GR1     LDA ##$3B
      0899  A0 C6                 LDY ##$C6
      089B  A2 7D                 LDX ##$7D
      089D  D0 06                 BNE SCREEN
      089F  A9 1B         GR0     LDA ##$1B
      08A1  A0 C7                 LDY ##$C7
      08A3  A2 15                 LDX ##$15         ; QUITA/PONE MODO BIT-MAP
      08A5  8D 11 D0      SCREEN  STA $D011         ; BANCO VIC (1 PARA GR#1)
      08A8  8C 00 DD              STY $DD00         ; MEMORIA PANTALLA Y COLOR
      08AB  8E 18 D0              STX $D018
      08AE  60            EXIT    RTS
      08AF                ;
      08AF                COLR
      08AF                ;
      08AF  20 EB B7              JSR INPUT2        ; ENTRA COLORES
      08B2  8A                    TXA               ; COLOR DE CARACTERES
      08B3  0A                    ASL A             ; DESPLAZA EL NYBBLE EN A
      08B4  0A                    ASL A             ; (NIBBLE=MEDIO BYTE)
      08B5  0A                    ASL A
      08B6  0A                    ASL A
      08B7  05 14                 ORA $14           ; COLOR DE FONDO
      08B9  A2 5C                 LDX ##$5C
      08BB  86 FC                 STX $FC           ; PUNTERO A MEMORIA DE COLOR
      08BD  A0 00                 LDY ##$00
      08BF  84 FB                 STY $FB
      08C1  91 FB         FILL    STA ($FB),Y       ; LLENA MEMORIA DE COLOR
      08C3  C8                    INY
      08C4  D0 FB                 BNE FILL
      08C6  E6 FC                 INC $FC
      08C8  A6 FC                 LDX $FC
      08CA  E0 60                 CPX ##$60         ; $6000=FINAL DE RAM DE COLOR
      08CC  90 F3                 BCC FILL
      08CE  60                    RTS
      08CF                ;
      08CF                LEFT
      08CF                ;
      08CF  20 9E AD              JSR EVALEX        ; ENTRA ANGULO
      08D2  A5 66                 LDA $66           ; CAMBIA SIGNO DEL ANGULO
      08D4  49 FF                 EOR ##$FF         ; GIRA DESDE 'LEFT 90'
      08D6  85 66                 STA $66           ; HASTA 'RIGHT 90'.
      08D8  50 03                 BVC RIGHT+3
      08DA                ;
      08DA                RIGHT
      08DA                ;
      08DA  20 9E AD              JSR EVALEX        ; ENTRA ANGULO
      08DD                        MULPLY DEGRAD     ; LO PASA A RADIANES
      08E4                        ADD CURANG        ; LE SUMA EL ACTUAL
```

EXCURSION POR LA MICROPROGRAMACION

Si el dibujo es demasiado alto, sustituye el valor por uno menor, como 0,65. Si tu dibujo es demasiado ancho, prueba otro valor como 0,8. Una vez que encuentres el valor que necesitas, puedes sustituirlos en el programa para que siempre se cargue con él. La sintaxis correcta de la función USR es A=USR(X), por ejemplo

A=USR(.8).

Cómo funciona

Dada la interesante estructura del Commodore 64, existen al menos tres modos de añadir nuevos comandos al Basic. Está el

```
LINE# LOC   CODE           LINE

                                   UPDATE CURANG
00202  08EB                        JSR COSINE
00209  08F2  20 64 E2              UPDATE DX      ; DX=COS(CURANG)
00210  08F5                        FETCH CURANG
00217  08FC                        JSR SINE
00224  0903  20 6B E2              MULPLY SQUARE
00225  0906                        XLOYHI DY      ; DY=SIN(CURANG)*SQUARE
00232  090D                        JMP FACMEN     ; ACTUALIZA DY
00236  0911  4C D7 BB       ;
00237  0914                 PLOT
00238  0914                 ;                     ; ENTRA COORDENADAS
00239  0914                        JSR INPUT2     ; ALMACENA YCOR
00240  0914  20 EB B7              STX $02        ; YCOR=199-YCOR: ESTO PONE
00241  0917  86 02                 LDA #$C7       ; EL ORIGEN DE LA PANTALLA
00242  0919  A9 C7                 SBC $02        ; ABAJO A LA IZQUIERDA.
00243  091B  E5 02                 STA $02        ; XCOR EN $14,$15
00244  091D  85 02                 LDA $15        ; SI XCOR ESTA FUERA DE PANTALLA
00245  091F  A5 15                 BEQ OK         ; NO LO DIBUJA.
00246  0921  F0 0B                 CMP #$01
00247  0923  C9 01                 BNE NOTOK
00248  0925  D0 06                 LDA $14
00249  0927  A5 14                 CMP #$40
00250  0929  C9 40                 BCC OK
00251  092B  90 01          NOTOK  RTS            ; SI YCOR ESTA FUERA DE
00252  092D  60             OK     LDA $02        ; PANTALLA NO LO DIBUJA.
00253  092E  A5 02                 CMP #$C8
00254  0930  C9 C8                 BCS NOTOK
00255  0932  B0 F9          ; PARA SELECCIONAR EL PUNTO CORRECTO
00256  0934                 ; SE UTILIZAN LAS SIGUIENTES ECUACIONES
00257  0934                 ; CHAR=BASE+40*INT(Y/8)+INT(X/8)
00258  0934                 ; BYTE=YAND7  ,    BIT=XAND7
00259  0934                        AND #$07       ; Y GUARDA EL BYTE
00260  0934  29 07                 TAY
00261  0936  A8                    LDA #$00
00262  0937  A9 00                 STA $FC
00263  0939  85 FC                 LDA $02
00264  093B  A5 02                 AND #$F8       ; INT (Y/8)
00265  093D  29 F8                 ROL A
00266  093F  2A                    ROL $FC
00267  0940  26 FC                 ROL A
00268  0942  2A                    ROL $FC
00269  0943  26 FC                 ROL A          ; INT(Y/8)*8
00270  0945  2A                    ROL $FC        ; GUARDA VALOR INTERMEDIO
00271  0946  26 FC                 STA $FB
00272  0948  85 FB                 LDX $FC
00273  094A  A6 FC                 ROL A
00274  094C  2A                    ROL $FC
00275  094D  26 FC                 ROL A          ; INT(Y/8)*32
00276  094F  2A                    ROL $FC        ; SUMA BYTE BAJO INT(Y/8)*8
00277  0950  26 FC                 ADC $FB
00278  0952  65 FB                 STA $FB
00279  0954  85 FB                 TXA            ; SUMA BYTE ALTO
00280  0956  8A                    ADC $FC        ; LE SUMA LA BASE ($6000)
00281  0957  65 FC                 ORA #$60
00282  0959  09 60                 TAX
00283  095B  AA
```

EXCURSION POR LA MICROPROGRAMACION

auténtico wedge o cuña, que desafortunadamente ralentiza la ejecución del intérprete basic (este método podría utilizarse en el VIC-20).

Un modo mejor es copiar la ROM a la RAM que ocupa las mismas direcciones y entonces modificar los comandos y vectores para satisfacer tus necesidades. El problema es que pierdes un comando de los existentes por cada uno nuevo. Comencé mi proyecto utilizando este método, pero decidí que no quería eliminar el uso de DEF FN o VERIFY, por citar sólo un par de comandos. De modo que

```
LOC   CODE           LINE

095C  A5 14                 LDA $14           ; INT(Y/8)
095E  29 F8                 AND #$F8          ; SUMA BYTE BAJO INT(X/8)
0960  65 FB                 ADC $FB
0962  85 FB                 STA $FB
0964  8A                    TXA               ; SUMA BYTE ALTO
0965  65 15                 ADC $15
0967  85 FC                 STA $FC
0969  A5 14                 LDA $14
096B  29 07                 AND #$07
096D  AA                    TAX               ; X GUARDA EL BIT
096E  B1 FB                 LDA ($FB),Y       ; SELECCIONA EL BYTE
0970  1D 1F 82              ORA BITVAL,X      ; ACTIVA EL BIT
0973  91 FB                 STA ($FB),Y
0975  60                    RTS
0976                 ;
0976                 DRAW
0976                 ;                        ; #$E6=AZUL CLARO SOBRE
0976  A9 E6                 LDA #$E6          ; FONDO OSCURO.
0978  85 97                 STA FLAG          ; BAJA LAPIZ.
097A  20 61 80              JSR COLOR
097D  A0 0F                 LDY #$0F
097F  B9 18 80       INITXY LDA STARTX-1,Y    ; FIJA XCOR,YCOR
0982  99 09 80              STA XCOR-1,Y      ; Y EL ANGULO
0985  88                    DEY
0986  D0 F7                 BNE INITXY
0988  84 FB                 STY $FB           ; PUNTERO A PANTALLA
098A  A9 60                 LDA #$60
098C  85 FC                 STA $FC
098E  98                    TYA
098F  91 FB          CLEAR  STA ($FB),Y       ; BORRA PANTALLA
0991  C8                    INY
0992  D0 FB                 BNE CLEAR
0994  E6 FC                 INC $FC
0996  10 F7                 BPL CLEAR         ; PONE CERO EN FAC#1
0998  20 91 B3              JSR FIXFLO        ; INICIALIZA DX Y DY
099B  20 85 80              JSR FINDDX        ; ACTIVA PANTALLA GRAFICA
099E  4C 3F 80              JMP IGR1
09A1                 ;
09A1                 TAIL
09A1                 ;                        ; UN CARACTER DISTINTO DE 'U'
09A1  E9 55                 SBC #'U           ; BAJARA EL LAPIZ.
09A3  85 97                 STA FLAG
09A5  20 73 00       IGNORE JSR CHRGET IGNORA EL PRIMER
09A8  D0 FB                 BNE IGNORE        ; CARACTER
09AA  60                    RTS
09AB                 ;
09AB                 MOVE
09AB                 ;                        ; ENTRA LONGITUD A MOVER
09AB  20 9E AD              JSR EVALEX
09AE  A5 66                 LDA $66           ; SI ES NEGATIVA VUELVE
09B0  30 69                 BMI DONE
09B2  20 9B BC              JSR FLOFIX
09B5  A6 65                 LDX $65
09B7  A5 64                 LDA $64           ; SI 'MOVE 511', X=#$FF
09B9  F0 03                 BEQ SHORT
```

EXCURSION POR LA MICROPROGRAMACION

escogí otro método.

Cuando tecleas una línea de Basic y pulsas Return, el intérprete basic evalúa la línea y la convierte en "tokens". Si el intérprete espera encontrar una palabra clave (comando o función) pero no lo encuentra, asume que se trata de un LET, y salta a ejecutar la rutina LET.

Hasta ahora es así. Cambiando este salto de modo que apunte a una nueva rutina de evaluación te permite añadir nuevas palabras clave aunque tú no las comprimirás (no las convertirás en tokens). Esta rutina no disminuye la velocidad de ejecución de

```
LINE# LOC · CODE        LINE

                        LDA #$FF        ; PERO SE INCREMENTA A #$00
09BB  A9 FF             INX             ; ES INSOSPECHABLE.
09BD  E8         SHORT  STA $19
09BE  85 19             STX $1A
09C0  86 1A      NEXT   FETCH XCOR      ; XCOR=XCOR+DX
09C2                    ADD DX
09C9                    UPDATE XCOR
09D0                    JSR FLOFIX
09D7  20 9B BC          LDX $65
09DA  A6 65             LDY $64         ; LO GUARDA PARA 'PLOT'
09DC  A4 64             STY $15
09DE  84 15             STX $14
09E0  86 14             FETCH YCOR      ; YCOR=YCOR+DY
09E2                    ADD DY
09E9                    UPDATE YCOR
09F0                    JSR FLOFIX      ; ELIMINA YCOR>255
09F7  20 9B BC          LDX $64
09FA  A6 64             BNE NEXT1
09FC  D0 0B             LDX $65
09FE  A6 65             STX $02         ; SI EL LAPIZ ESTA LEVANTADO
0A00  86 02             LDA FLAG        ; NO PINTA
0A02  A5 97             BEQ NEXT1
0A04  F0 03             JSR PLOT1
0A06  20 C7 80   NEXT1  LDA $19
0A09  A5 19             BEQ NEXT2
0A0B  F0 05             DEC $19
0A0D  C6 19             CLC
0A0F  18         STEP   BCC NEXT
0A10  90 B0      NEXT2  LDA $1A
0A12  A5 1A             BEQ DONE
0A14  F0 05             DEC $1A
0A16  C6 1A             CLC
0A18  18                BCC STEP
0A19  90 F5      DONE   RTS
0A1B  60         ;
0A1C             ERROR
0A1C             ;
0A1C                    TXA
0A1C  8A                BMI NOERR       ; VUELVE A PANTALLA DE TEXTO
0A1D  30 08             PHA             ; PARA VISUALIZAR
0A1F  48                JSR IGR0        ; MENSAJE DE ERROR.
0A20  20 47 80          PLA
0A23  68                JMP IERROR
0A24  4C 3B A4   NOERR  JMP READY
0A27  4C 74 A4   ;
0A2A             USR
0A2A             ;
0A2A                    UPDATE SQUARE   ; NUEVO VALOR DE CUADRO
0A2A                    JMP FINDDY      ; EL DE 'DY'
0A31  4C A4 80   ;
0A34             PARSER
0A34             ;
0A34                    LDX #$00
0A34  A2 00             LDY #$00
0A36  A0 00      SEARCH LDA ($7A),Y
0A38  B1 7A
```

EXCURSION POR LA MICROPROGRAMACION

las palabras basic normales exceptuando LET. Pero asumiendo que LET "viaja" ya por un tortuoso camino (perdiéndose por 35 palabras), la adición de nuevas palabras es sólo un pequeño desvío.

Si en cualquier momento tu ordenador rechazase los nuevos comandos del Turtle Graphics dándote "Syntax Error", teclea POKE 1,54 para volver a habilitarlos. Esto no es necesario después de pulsar STOP y RESTORE.

Necesitarás un monitor de lenguaje máquina o un ensamblador para entrar este programa en tu ordenador. El listado 1 está escrito en lenguaje ensamblador; el listado 2 es un volcado hexadecimal de lenguaje máquina. Cuan-

```
LINE# LOC   CODE          LINE

00436  0A3A  DD 27 82            CMP TABLE,X     ; LO COMPARA CON LAS INICIALES
00437  0A3D  F0 0C               BEQ MATCH       ; DE LA TABLA.
00438  0A3F  8A                  TXA             ; NO CONCUERDA
00439  0A40  18         MISSED   CLC             ; X APUNTA A LA PROXIMA ENTRADA
00440  0A41  69 08               ADC #$08
00441  0A43  AA                  TAX
00442  0A44  C9 40               CMP #$40        ; ES EL FINAL DE LA TABLA?
00443  0A46  D0 F0               BNE SEARCH
00444  0A48  4C A5 A9            JMP LET         ; SI LO ES SALTA A RUTINA LET
00445  0A4B  C8         MATCH    INY
00446  0A4C  E8                  INX
00447  0A4D  B1 7A               LDA ($7A),Y     ; COMPARA EL RESTO DE
00448  0A4F  DD 27 82            CMP TABLE,X     ; LA PALABRA CON LA TABLA.
00449  0A52  F0 F7               BEQ MATCH
00450  0A54  09 80               ORA #$80        ; ULTIMO CARACTER EN TABLA?
00451  0A56  DD 27 82            CMP TABLE,X
00452  0A59  F0 07               BEQ FOUND
00453  0A5B  8A                  TXA
00454  0A5C  29 F8               AND #$F8
00455  0A5E  A0 00               LDY #$00        ; INTENTARLO OTRA VEZ
00456  0A60  F0 DE               BEQ MISSED
00457  0A62  BD 28 82   FOUND    LDA TABLE+1,X   ; PONE VECTOR EN STACK
00458  0A65  48                  PHA
00459  0A66  BD 29 82            LDA TABLE+2,X
00460  0A69  48                  PHA
00461  0A6A  8A                  TXA
00462  0A6B  29 07               AND #$07        ; LONGITUD DE LA PALABRA
00463  0A6D  AA                  TAX
00464  0A6E  20 73 00   MOVEP    JSR CHRGET      ; AVANZA PUNTERO
00465  0A71  CA                  DEX
00466  0A72  D0 FA               BNE MOVEP
00467  0A74  4C 73 00            JMP CHRGET      ; SALTA A LA RUTINA
00468  0A77                   ; POR MEDIO DE CHRGET.
00469  0A77                   ;
00470  0A77                   ; BITVAL
00471  0A77                   ;
00472  0A77  80                  .BYTE $80,$40,$20,$10,$08,$04,$02,$01
00472  0A78  40
00472  0A79  20
00472  0A7A  10
00472  0A7B  08
00472  0A7C  04
00472  0A7D  02
00472  0A7E  01
00473  0A7F                   ;
00474  0A7F                   ; TABLA DE PALABRAS Y VECTORES
00475  0A7F                   ;
00476  0A7F  54 41 49            .BYTE 'TAI',$CC,$81,$48,$00,$00
00476  0A82  CC
00476  0A83  81
00476  0A84  48
00476  0A85  00
00476  0A86  00
00477  0A87  52 49                .BYTE 'RIGH',$D4,$80,$81,$00
00477  0A8B  D4
```

EXCURSION POR LA MICROPROGRAMACION

do termines de entrar el programa, guárdalo en cinta o disco (antes de probarlo), empezando en la dirección hexadecimal $0801 (en lugar de $0800), de este modo podrás cargarlo y ejecutarlo con LOAD y RUN como un programa en Basic normal.

El programa de demostración (listado 3) te dará algunas ideas de cómo utilizar los comandos del Turtle Graphics. Observa cómo se dibujan los círculos. Verás, que en muchos casos, cambiando solamente un número

```
LINE# LOC   CODE          LINE

00477  0A8C  80
00477  0A8D  81
00477  0A8E  00
00478  0A8F  50 4C 4F            .BYTE 'PLO',$D4,$80,$BB,$00,$00
00478  0A92  D4
00478  0A93  80
00478  0A94  BB
00478  0A95  00
00478  0A96  00
00479  0A97  4D 4F 56            .BYTE 'MOV',$C5,$81,$52,$00,$00
00479  0A9A  C5
00479  0A9B  81
00479  0A9C  52
00479  0A9D  00
00479  0A9E  00
00480  0A9F  4C 45 46            .BYTE 'LEF',$D4,$80,$76,$00,$00
00480  0AA2  D4
00480  0AA3  80
00480  0AA4  76
00480  0AA5  00
00480  0AA6  00
00481  0AA7  47 52               .BYTE 'GR',$A3,$80,$31,$00,$00,$00
00481  0AA9  A3
00481  0AAA  80
00481  0AAB  31
00481  0AAC  00
00481  0AAD  00
00481  0AAE  00
00482  0AAF  44 52 41            .BYTE 'DRA',$D7,$81,$1D,$00,$00
00482  0AB2  D7
00482  0AB3  81
00482  0AB4  1D
00482  0AB5  00
00482  0AB6  00
00483  0AB7  43 4F 4C            .BYTE 'COL',$D2,$80,$56,$00,$00
00483  0ABA  D2
00483  0ABB  80
00483  0ABC  56
00483  0ABD  00
00483  0ABE  00
00484  0ABF                      .END

ERRORS = 00000

SYMBOL TABLE

SYMBOL VALUE
ADD      FFFF   ALOYHI  FFFF   BASIC   0801   BITVAL  821F
CHRGET   0073   CLEAR   098F   COLOR   8061   COLR    08AF
COSINE   E264   CURANG  8014   DEGRAD  8028   DONE    0A1B
DRAW     0976   DX      8000   DY      8005   ERROR   0A1C
EVALEX   AD9E   EXIT    08AE   FACMEN  BBD7   FETCH   FFFF
```

EXCURSION POR LA MICROPROGRAMACION

obtenemos un dibujo totalmente diferente.

Como dije al principio, es interesante que te fijes en la utilización de los macros, por ejemplo cómo se define MULPLY en función de otro marco (ALOYHI) con el parámetro ?1 y luego es utilizado con una macroinstrucción como MULPLY DEGRAD.

En otros ejemplos intentaré equilibrar el VIC para que no se sientan menospreciados sus usuarios.

```
SYMBOL TABLE

SYMBOL VALUE
 FILL     08C1    FINDDX   8085    FINDDY   80A4    FIXFLO   B391
 FLAG     0097    FLOFIX   BC9B    FOUND    0A62    GR       088A
 GR0      089F    GR1      0897    IERROR   A43B    IGNORE   09A5
 IGR0     8047    IGR1     803F    INIT     080E    INITXY   097F
 INPUT2   B7EB    LEFT     08CF    LET      A9A5    LOOP1    081A
 LOOP2    0855    MATCH    0A4B    MEMFAC   BBA2    MISSED   0A40
 MOVE     09AB    MOVEP    0A6E    MULPLY   FFFF    MULT     BA28
 NEXT     09C2    NEXT1    0A09    NEXT2    0A12    NOERR    0A27
 NOTOK    092D    OK       092E    PARSER   0A34    PLOT     0914
 PLOT1    80C7    PLUS     B867    READY    A474    RIGHT    08DA
 SCREEN   08A5    SEARCH   0A38    SHORT    09BE    SINE     E26B
 SOURCE   0871    SQUARE   802D    STARTA   8023    STARTX   8019
 STARTY   801E    STEP     0A10    TABLE    8227    TAIL     09A1
 UPDATE   FFFF    USR      0A2A    XCOR     800A    XLOYHI   FFFF
 YCOR     800F

END OF ASSEMBLY
```

```
LISTADO 2

.:0800 00 0C 08 0A 00 9E 20 32
.:0808 30 36 32 00 00 00 A9 37
.:0810 85 01 A9 A0 85 FC A0 00
.:0818 84 FB B1 FB 91 FB C8 D0
.:0820 F9 E6 FC 24 FC 50 F3 A9
.:0828 36 85 01 A9 81 8D 01 03
.:0830 8D 12 03 8D 06 A8 A9 C4
.:0838 8D 00 03 A9 D2 8D 11 03
.:0840 A9 DC 8D 05 A8 A9 80 85
.:0848 FE A9 19 85 FD A9 08 85
.:0850 FC A9 71 85 FB B1 FB 91
.:0858 FD C8 D0 F9 E6 FE E6 FC
.:0860 A5 FC C9 0B 90 EF A2 00
.:0868 A0 5C 18 20 99 FF 4C 02
.:0870 E4 88 1F 00 00 00 87 46
.:0878 00 00 00 83 16 CB E3 F8
.:0880 7B 0E FA 35 12 80 40 00
.:0888 00 00 48 20 73 00 68 C9
.:0890 30 F0 0C C9 31 D0 17 A9
.:0898 3B A0 C6 A2 7D D0 06 A9
.:08A0 1B A0 C7 A2 15 8D 11 D0
.:08A8 8C 00 DD 8E 18 D0 60 20
.:08B0 EB B7 8A 0A 0A 0A 0A 05
.:08B8 14 A2 5C 86 FC A0 00 84
.:08C0 FB 91 FB C8 D0 FB E6 FC
.:08C8 A6 FC E0 60 90 F3 60 20
.:08D0 9E AD A5 66 49 FF 85 66
.:08D8 50 03 20 9E AD A9 28 A0
.:08E0 80 20 28 BA A9 14 A0 80
.:08E8 20 67 B8 A2 14 A0 80 20
.:08F0 D7 BB 20 64 E2 A2 00 A0
.:08F8 80 20 D7 BB A9 14 A0 80
.:0900 20 A2 BB 20 6B E2 A9 2D
.:0908 A0 80 20 28 BA A2 05 A0
.:0910 80 4C D7 BB 20 EB B7 86
.:0918 02 A9 C7 E5 02 85 02 A5
.:0920 15 F0 0B C9 01 D0 06 A5
```

```
.:0928 14 C9 40 90 01 60 A5 02
.:0930 C9 C8 B0 F9 29 07 A8 A9
.:0938 00 85 FC A5 02 29 F8 2A
.:0940 26 FC 2A 26 FC 2A 26 FC
.:0948 85 FB A6 FC 2A 26 FC 2A
.:0950 26 FC 65 FB 85 FB 8A 65
.:0958 FC 09 60 AA A5 14 29 F8
.:0960 65 FB 85 FB 8A 65 15 85
.:0968 FC A5 14 29 07 AA B1 FB
.:0970 1D 1F 82 91 FB 60 A9 E6
.:0978 85 97 20 61 80 A0 0F B9
.:0980 18 80 99 09 80 88 D0 F7
.:0988 84 FB A9 60 85 FC 98 91
.:0990 FB C8 D0 FB E6 FC 10 F7
.:0998 20 91 B3 20 85 80 4C 3F
.:09A0 80 E9 55 85 97 20 73 00
.:09A8 D0 FB 60 20 9E AD A5 66
.:09B0 30 69 20 9B BC A6 65 A5
.:09B8 64 F0 03 A9 FF E8 85 19
.:09C0 86 1A A9 0A A0 80 20 A2
.:09C8 BB A9 00 A0 80 20 67 B8
.:09D0 A2 0A A0 80 20 D7 BB 20
.:09D8 9B BC A6 65 A4 64 84 15
.:09E0 86 14 A9 0F A0 80 20 A2
.:09E8 BB A9 05 A0 80 20 67 B8
.:09F0 A2 0F A0 80 20 D7 BB 20
.:09F8 9B BC A6 64 D0 0B A6 65
.:0A00 86 02 A5 97 F0 03 20 C7
.:0A08 80 A5 19 F0 05 C6 19 18
.:0A10 90 B0 A5 1A F0 05 C6 1A
.:0A18 18 90 F5 60 8A 30 08 48
.:0A20 20 47 80 68 4C 3B A4 4C
.:0A28 74 A4 A2 2D A0 80 20 D7
.:0A30 BB 4C A4 80 A2 00 A0 00
.:0A38 B1 7A DD 27 82 F0 0C 8A
.:0A40 18 69 08 AA C9 40 D0 F0
.:0A48 4C A5 A9 C8 E8 B1 7A DD
.:0A50 27 82 F0 F7 09 80 DD 27
.:0A58 82 F0 07 8A 29 F8 A0 00
```

EXCURSION POR LA MICROPROGRAMACION

```
.:0A60 F0 DE BD 28 82 48 BD 29
.:0A68 82 48 8A 29 07 AA 20 73
.:0A70 00 CA D0 FA 4C 73 00 80
.:0A78 40 20 10 08 04 02 01 54
.:0A80 41 49 CC 81 48 00 00 52
.:0A88 49 47 48 D4 80 81 00 50
.:0A90 4C 4F D4 80 BB 00 00 4D
.:0A98 4F 56 C5 81 52 00 00 4C
.:0AA0 45 46 D4 80 76 00 00 47
.:0AA8 52 A3 80 31 00 00 00 44
.:0AB0 52 41 D7 81 1D 00 00 43
.:0AB8 4F 4C D2 80 56 00 00 DF
```

LISTADO 3

```
1 REM DEMOSTRACION DEL TURTLE
10 DRAW:COLR14,6:TAILUP:RIGHT150:MOV
E150::LEFT150:TAILDOWN
20 FORG=1TO36:FORT=1TO2:FORR=1TO8:MO
VE9:LEFT10:NEXT
```

```
30 FORR=1TO6:MOVE9:RIGHT10:NEXTR,T:R
IGHT170:NEXT:GOSUB300
40 FORT=4TO8STEP2:A=360/T:DRAW
50 FORR=1TOT:FORG=1TOT:MOVEA:LEFTA:N
EXT:RIGHTA:NEXT
60 GOSUB300:NEXT
70 DRAW:POKE53280,0:COLR0,2:A=160:FO
RT=1TO200:MOVET:LEFTA:NEXT:GOSUB300
80 FORT=13TO15STEP2:A=720/T:DRAW
90 FORR=1TOT:FORG=1TOT:TAILDOWN:IFG/
2=INT(G/2)THENTAILUP
100 MOVEA:LEFTA:NEXT:RIGHTA:NEXT
110 GOSUB300:NEXT
120 DRAW:POKE53280,0:COLR0,6:FORT=1T
O24:FORR=1TO36
130 MOVE10:LEFT10:NEXT:RIGHT15:NEXT:
GOSUB300
140 DRAW:COLR6,8:A=89.5:FORT=1TO230S
TEP1.5:MOVET:LEFTA:NEXT:GOSUB300
299 GR#0:END
300 FORY=1TO1000:NEXT:POKE53280,14:R
ETURN
```

COMENTARIOS COMMODORE WORLD

(Viene de pág. 51)

La cuarta sección, titulada "Más sobre programación", da más detalles sobre la utilización de técnicas de programación avanzadas, ya que la segunda sección era muy elemental y no detallaba todas las posibilidades del Commodore 64. En esta sección del libro encontramos capítulos dedicados a instrucciones condicionales (IF/THEN), cómo utilizar las sentencias DATA/READ, cómo poder adaptar los programas que deseemos a nuestras necesidades particulares, y una serie de programas algo más complejos que los que habíamos encontrado hasta este momento.

Las cuatro secciones anteriores constan de un total de dieciocho capítulos, que cumplen perfectamente su función: servir de guía para el usuario durante las primeras semanas o meses de utilización del ordenador, y no decírselo todo sobre el, ya que esto último supondría utilizar un manual de más de mil páginas y aún quedarían puntos sin aclarar. La finalidad del libro la deja bien clara el autor en la introducción, en la que dice que tiene un alcance limitado. Creemos que aunque esté limitado, mucho más lo están los manuales y dejan demasiados puntos sin aclarar, por lo que esta obra es muy útil para todos los principiantes.

Para terminar el estudio, encontramos cincuenta y tres páginas destinadas a los apéndices. Los temas tratados en esta parte del libro podrían confundir el estudio inicial del ordenador, por lo que pueden verse al repasar posteriormente lo aprendido. Describen todas las peculiaridades y complejidades del ordenador personal C-64, y son: El comando Poke, la instrucción Peek, los sprites, los sonidos, qué hacer con sus propios programas, una descripción de todas las palabras clave del lenguaje basic, y un apéndice dedicado a las tablas de códigos de pantalla, ASCII, mapas de memoria de pantalla y color, valores para obtener las notas musicales, mapas de los registros de sprites y bases del control de sonido del Commodore 64.

MARKET CLUB

Servicio gratuito para nuestros lectores *PARTICULARES*. Los anuncios serán publicados durante 1 mes. Debido a ciertos abusos que nos han sido comunicados, los anuncios gratuitos de Market Club *SOLAMENTE* serán publicados si vienen con nombre, apellidos y dirección completa.

MERCADILLO

• Compro Controlador C8 de 8 reles (precio a convenir) o lo cambio por cartucho SUPEREXPANDER +3K para el VIC-20 y un cassette con juegos y programas también para el VIC-20. Vendo libro "INTRODUCCION AL LENGUAJE DE PROGRAMACION BASIC" para el VIC por 1.000 ptas., cartucho SUPEREXPANDER +3K por 7.000 ptas. (todo por 9.000 ptas.) (VIC 20). Me interesaría ponerme en contacto con alguna persona que tuviera todos los números de CLUB COMMODORE o con alguien que tuviera esquemas de algun modem telefónico o de algún RTTY-CW para CBM 64. Llamar de 19,30 a 23 h. a José de la Fuente, al Telf. (93) 242 51 27. 08015 Barcelona. ***(Ref. M-174).***

• Desearía comprar instrucciones sobre el manejo y programación del Commodore 64 en castellano. Carlos Iglesias, C/La Llacuna, 4. Andorra la Vella. (Principado de Andorra). ***(Ref. M-175)***

• Hola soy un VIC-cioso de Barcelona y buscaría alguien que me vendiera el Super Expander del Vic-20. Precio a convenir. Llamar al teléfono 794 07 09. O escribir al Paseo de la Misericordia nº 18 de Canet de Mar (Barcelona). Si llamáis preguntad por Lluis Maria. Llamad preferiblemente festivos o laborables de 19 a 23 h. ***(Ref. M-176).***

• Desearía comprar VIC-20 y Commodore 64, 1 Impresora tracción y otra tracción y fricción adaptable a los dos ordenadores, CARLES BONJOCH C/Mosen Amic nº 1. Pral. Lérida (973) 24 21 08 ***(Ref. M-177).***

• Angel Jiménez busca una fotocopia del Nº 1 agotado. Quien pueda suministrársela que se ponga en contacto con él: Angel Jiménez Camino. Avda. de Barcelona, 12-2º 1. Tels. 25-5643. Córdoba. ***(Ref. M-178).***

• Vendo para VIC-20 cassette CN2 —9.000 pts.—, memoria de 8K a 16K Ram por 11.000 ptas. cartucho de superexpander con 3K de memoria Ram 7.000 ptas. cartucho de monitor de código máquina 4.000 ptas. Jesús Prieto. Calle Barratón, 25. 28011 Madrid. Tel.: 463 72 30. ***(Ref. M-179).***

• Busco programas sobre un VIC-20 + 64K RAM sobre arquitectura, cálculo estructuras, pórticos, etc. Gratifico información. Máximo Martín. C/Orillamar, 75-4º izda. Tel. 204 460. 15002 La Coruña ***(Ref. M-180).***

• Vendo para VIC-20 cartucho de juegos: VIC-AVENGER referencia (1901) por 4.000 ptas. o lo cambiaria por el libro "Acceso rápido al VIC-20 (Getting acquainted with your VIC-20)". Ofertas llamar al (958) 272217, preguntar por: Julián (hijo) Manglaro García, C/Acera de Canasteros, 16, 4º A. 18012 Granada. Llamar de 10 a 11 de la noche ***(Ref. M-181).***

• Vendo VIC-20 nuevo por 24.000 ptas., regalo 2 libros de Basic curso de introducción al Basic en cinta (1.ª parte), y unos juegos. Intercambio programas en disco o cassette para C-64. Interesados dirigirse a: Blas Sánchez Navarro, C/Cruzadas, 58. Sabadell. Barcelona. Tel. 711 00 49 ***(Ref. M-182)***

• Intercambio programas para el C-64. Francesc Verdú. C/Barcelona, 65, 5º. 17001 Girona. Tel. (972) 201663 ***(Ref. M-183).***

• Vendo VIC-20 en perfecto estado (12-83) por 25.000 pesetas más cassette C2N (10.000 ptas.) más guía del Usuario del VIC (1.000 ptas) más dos cursos Introducción al Basic (1.500 ptas.) todo junto por 39.000 ptas., también por separado. Regalo libro "Acceso rápido al VIC-20". Manuel Aranda Atienza. C/Rio Sella, 10 - 4º B. Móstoles - (Madrid). Tel.: 617 97 31. ***(Ref. M-184).***

• Poseo un C-64 con el cassette C2N y deseo intercambiar programas de utilidad en general, y programas relacionados con la radioafición. Interesados dirigirse al P.O. Box 1, Peñaflor (Sevilla) o llamar de 21 a 23 hras. de lunes a viernes al 807034 Antonio Cano Quesada. C/Granada, 34. Peñaflor (Sevilla). ***(Ref. M-185).***

• Cambio programa MASTER-64 por cartucho lenguaje Forth o vendo por 18.000 ptas. que también permite correr programas de las series 4.000 y 8.000 en el C-64. Fernando Franco Franco, C/Felipe de Paz, 12, 15º 1ª 08028 Barcelona. Tel.: 339 87 95. ***(Ref. M-186).***

• Vendo Commodore 64 con cassette, cartucho de Fútbol y Joystick, nuevo con garantía a precio interesante. Llamar a: Magin Borrell, San Carlos, 23. Badalona. Barcelona. Teléf. (93) 389 43 08. ***(Ref. M-187).***

• Vendo C-64 con cassette 65.000 ptas. José García González. C/Candoso, 2-2º F. Torrejón de Ardoz-Madrid. Tel. (91) 676 84 09. ***(Ref. M-188).***

• Intercambio programas del C-64. Vendo curso de introducción al C-64. I (1.500 ptas.) José Cortes A. C/Aurora, 19-4º 1.ª. 08001 Barcelona. Tel. 3298631. ***(Ref. M-189).***

• Desearía cambiar programas de Juegos u otras aplicaciones para C-64. Interesados llamar Teléf. (93) 432 16 87 o escribir a Eduardo Melero Vázquez C/Mármol, 3, 3º 2.ª. 08004 Barcelona ***(Ref. M-190).***

• Vendo ordenador VIC-20 con ampliación universal de 64K, 32 libres y 32 ocultas, datassette, joystik y tres juegos, todo por 55.000 ptas. Vendo ampliación de 16 K por 11.500 ptas. Andres Lacrur Limiñana. C/Urgel, 158, 4º piso. 08036 Barcelona. ***(Ref. M-191).***

• Vendo calculadora CASIO PB300 programable en basic. Sin estrenar. Llamar por las tardes al teléfono (942) 891106 ó escribir a Isabel Rodríguez Lara. Plaza 3 de Noviembre, nº 1-6º C. Torrelavega (Cantabria) ***(Ref. M-192).***

• Cambio cualquier tipo de programas y experiencia para el C-64. Sergi Arbós Mauri C/Forn del Vidre 16, 3.ª 4.ª Vilanova i la Geltrú (Barcelona). Tel. 893 18 31. ***(Ref. M-193).***

• Vendo C-64, además regalo, el datassette de Commodore, algunas revistas, y el manual en castellano de la Commodore. **Todo completamente a estrenar por ti.** Precio 70.000 (Valor en tienda 92.000 ±) Negociables en cierta manera. Teléf. 889 35 63. (Hora de comidas). Angel Antonio Corral Gutiérrez. C/Parque Falcón, nº 1. Alcalá de Henares (Madrid). ***(Ref. M-194)***

• **Vendo VIC-20 con datassette por 29.900 ptas.** Todo funciona de maravilla. El VIC-20 es de octubre del 83 y el datassette de enero del 84. Llamar al 357 04 29 de Barcelona. Preguntar por Germán Costa Aixas, C/Agudells, 56. 08032 Barcelona.- ***(Ref. M-195).***

• Intercambio juegos para el COMMODORE 64, todos en cinta, también tengo programas de aplicaciones. A los que les interese llamen al número 229 07 42 y pregunten por Francesc. Si les hace más ilusión escribirme, escriban a: Francesc Guasch Ortiz. Calle Jerez, 40 Torre, 08032 Barcelona. ***(Ref. M-196).***

• Vendo revistas de Club Commodore nº 3 y 4, por 200 ptas. cada una más gastos de envío (no son fotocopiadas). Teléf.: (972) 21 82 24, preguntar por Victor Roca. C/Maluquer Salvador, 24, 3º 3.ª. 17002 Girona. ***(Ref. M-197).***

• Cambio o vendo, el siguiente material VIC-20, cassette 3K+ superexpander, ampliación de 16K, cartucho, sargón II, Chess, más máquina fotográfica reflex para estreno, por Commodore 64, al mejor postor. (Ramón Romero Garcia Tel. 891 25 90, Doctor Fleming, 1. 4º Izda. Aranjuez) ***(Ref. M-198).***

• Vendo C-64 con cassette commodore comprado en Febrero del 84 suscripción a la revista desde el número 1 y muchos programas por 75.000 ptas. Francisco Rodríguez González, 222 380 ó 255 635 de Badajoz. Avd. Juan Sebastián Elcano, 33, 6º B ***(Ref. M-190).***

• Vendo VIC-20, cassette C2N, guía de referencia del programador y revistas Commodore". José Antonio Serrano Bergali. C/Avda. de Kansas City, 32. Tel. (954) 57 33 08. 41007 Sevilla. ***(Ref. M-200).***

• Se vende por cambio de ordenador: 1 ordenador VIC-20; 1 ampliación de memoria de 3K, 8K, 16K, siendo opcional conectar los 3, 8 ó 16K; 2 vídeo juegos en cartucho; 3 cintas de juegos; 1 programa en cinta para el VIC-20 de 40 columnas; 1 cinta con juegos; 1 guía de referencia del programador; Los dos cursos de introducción al basic, más las cintas. José M. Framit Campaña, C/Concepción Arenal, nº 1, 4º Izda. Telf.: 958 - 28 50 04. Granada 18012 ***(Ref. M-201).***

• Vendo VIC-20, con guía de referencia, lenguaje de programación Basic, con las dos cintas y un montón de juegos. Todo por 22.000 pesetas. Llamar al 333 09 41, José Martín Torres, C/Desamparados nº 22, 3º piso. Barcelona ***(Ref. M-202).***

• Vendo VIC-20 (diciembre 83), más datassette, más Joystick (model 1311), por 35.000 ptas. Regalo muchos programas y juegos y 10 cintas vírgenes de 60 min. y mucha información y revistas. Daniel Bassas Pablo. Pl. Marqués de Camps, nº 16, 3º 2.ª 17001 Girona. Tel.: (972) 212 726. ***(Ref. M-203).***

• Si alguien está interesado en obtener a bajo precio, alguno de los volúmenes de "INTRODUCCION AL BASIC" (I y II) o el "Manual de Referencia del programador", le ruego se ponga en contacto conmigo lo antes posible. Interesados ponerse en contacto conmigo: Patricio Blanco C/Can Noguera, nº 52, 2º B. La Garriga (Barcelona). ***(Ref. M-204).***

• Vendo Commodore 64, Unidad de cassette, introducción al basic del VIC-20 la unidad de disco VC-1541. Todo está en perfecto estado, ya que cambio de ordenador, lo vendería todo por 100.000 pesetas. César Pascual. Estación Renfe, s/n. Manresa. Tel. 872 26 50. ***(Ref. M-205).***

CLUBS

• Club de amigos. Foto Estudio 2. Plaza de Sombrereros, 2. **Palma de Mallorca**. Tel.: 21 31 62. ***(Ref. C-1).***

• Club de programación Alaiz. Tels.: 254 480-257 704. Pamplona. ***(Ref. C-4)***

• Club de Commodore de Albacete. Fernando Martínez Guerrero. La Roda, 39, 5º D. Albacete. ***(Ref. C-5)***

• Círculo de Durensan Vigués. Vigo. C/Venezuela, 48. Entrechan. Tels.: (986) 410 683/422 519. ***(Ref. C-6)***

• Asociación Manchega de Usuarios de CBM-64 (A.M.U. C-64), C/Blasco de Garay, 10. Albacete ***(Ref. C-13)***

• Club MICRO-ESPLAI. Gran Via , 682. Pral. (Entre Bruch y Gerona) de Barcelona. ***(Ref. C-14).***

• Interesados en la formación de un Club de informática en Granada capital, diríjanse por correo a C/San Jerónimo, 23-2º, 18001 Granada. Miguel A. Puerta Carrasco. ***(Ref. C-28).***

• Si queréis que hagamos un club "Vic-20" poneros en contacto con Antonio Recober. Teléf.: 32 20 46 Málaga. Si algún 64 quiere hacer el camino a nuestro lado, podemos intentarlo. ***(Ref. C-29).***

• Desearía contactar con interesados en formar un club de usuarios del C-64 en Granada. Antonio Toral Peinado. C/Puente nº 3. Illora (Granada). Tel. 958/46 30 09. ***(Ref. M-30).***

• Para formación club nacional de CBM Series 2000 y 3000 exclusivamente, ponerse en contacto por carta con Artemio González Pérez. Fernando Arocena Quintero, 1. Teléf.: 22 42 44. 38009 Santa Cruz de Tenerife. ***(Ref. C-31).***

• Me gustaría contactar con alguien de Motril para compartir experiencias y/o formar un club de C-64 (y VIC-20 si hace falta). Ya tengo local. Mi teléfono: 958-600 314 ***(Ref. C-32).***

• Ha nacido el club "Brian & Chip" para usuarios del C-64 y Spectrum 16K. Crearemos una revista propia. Intercambiamos programas listados o en cinta, escribir desde cualquier rincón de España. Francisco Reig López, C/Marina, 261, 3º 5.ª, 08025 Barcelona. ***(Ref. C-33).***

• Agradecería la dirección o teléfono de algún socio de mi ciudad o alrededores, a fin de ponerme en contacto con él para ver la posibilidad de formar un club o similar, José Esteban Llusar Molés. Daoiz, 2-1º 2.ª Pto. Sagunto (Valencia). ***(Ref. C-34).***

• Grupo de usuarios de Commodore en Tarra-

gona, estamos intentando formar un club lo más sólido posible, no nos mueve ningún interés lucrativo. Alfred Blasi. Tel.: (977) 547 307. Tarragona. *(Ref. C-35).*

• Deseo contactar con niños-as que deseen formar un club de ordenadores Commodore 64 y VIC-20 C/Avda. del Norte nº 8 atc. 3.ª Hospitalet (Barcelona). Manuel González Martínez. Tel.: 93-240 23 50. *(Ref. C-36)*

• Estamos montando una actividad de informática en el Club Los Chopos para chavales, sin ningún fin lucrativo. Javier Huguet, C/Antonio Menchaca, 19. Teléf. 469 34 61. Neguri. *(Ref. C-38).*

• Me gustaría contactar con usuarios de la zona de Mataró. Dirigirse a: Marcel Planagumá. Pge. Mir i Borrell, 1-1º 2.ª Mataró (Barcelona). Tel. (93) 7985127 *(Ref. C-42).*

• Desearía información sobre clubs en existencia y gente que quiera formar uno en Pontevedra de Commodore 64. Carlos Parada Gandos. C/Padre Fernando Olmedo, 4-4º B. 36002 Pontevedra. Tel. (986) 85 69 21. *(Ref. C-43).*

• Ha nacido el Club de Informática y videojuegos, el que nos escriba recibirá su carnet, con nº, intentaremos crear un presidente por ciudad (el 1º en escribir de cada una). Juan A. Manso. C/Illescas, 81. 28024 Madrid *(Ref. C-44)*

• Zaragoza. Club de amigos del VIC-20 y C-64 en ESPUMOSOS los sábados de 9,30 a 12,30. Daniel Fdez. de Velasco, C/Gran Via, 39. Zaragoza. *(Ref. C-45).*

• Les ruego si puede ser que me pongan en contacto con algún usuario del VIC-20. Daniel Roig. C/Castilla, 27. Ibiza (Baleares) *(Ref. C-46).*

• Me gustaría contactar con alguien de **Figueras** o alrededores para intercambiar programas o formar un Club Commodore 64. Josep Teixidor Planas, Vilasacra, 2. El Far D, Emporda (Girona). Tel. 508649. *(Ref. C-47).*

• VIC-20 Desearía un intercambio de experiencias en cuanto a programación, aplicaciones al campo de la radioafición (intefaces) y programas diversos. EC 6 MF. Lorenzo Sabater Coll. C/Agua, 2. Esporlas, Mallorca, Baleares. *(Ref. C-48).*

• Estamos formando un club de usuarios del COMMODORE 64 en Las Palmas. Desearíamos contactar con chicos y chicas para intercambio de programas y experiencias. Interesados dirigirse a: César Montenegro Armas, C/Pío XII, 56 2-D. Tel. 24 60 61. *(Ref. C-49).*

• Desearía ponerme en contacto con usuarios del VIC-20 para intercambiar programas e información. Carles Balañá. C/Tívoli, nº 11, 5ºA. Réus (Tarragona). *(Ref. C-50).*

• Som dos nois de SABADELL que estem interassats en formar un club d'usuaris d'ordenadors COMMODORE i SINCLAIR. Els interassats o interassades, poden escriure a: Daniel Ribes Garolera, C/Via Aurelia nº 24, 2º 3º Sabadell. Tambe demanen si es posible informació sobre clubs ja existens. *(Ref. C-51).*

• Desearía contactar por correo con usuarios del VIC-20, tanto usuarios nacionales como extranjeros. Escribir a Mª Mercedes a la C/Marqués del Muni, 25 (Guía de Gran Canaria). *(Ref. C-52).*

• Se ha creado "COESPECTRUM 16", un nuevo Club, nacido para todos los usuarios de CBM 64, Spectrum 16K y Dragon 32 y 64. Pero para que todo ello se haga realidad, necesitamos colaboradores y muchas ideas. Se intercambiarán toda clase de programas o listados. Se creará una revista. Escribir a Avd. de la Aurora, 57, 12º A. (29006) Málaga, desde cualquier punto de España, se contestará o llamar a: Angel, Tel. 321799, o Emilio, Tel. 398169. *(Ref. C-53).*

"Commodore World"
aparte de venderse generalmente en kioscos, se encuentra asimismo a la venta en las siguientes distribuciones de Commodore y librerías.

ALICANTE
- Casa Wagner, c/Juan Carlos I, 37. Elda. Teléfono: (965) 39 03 96
- Ferretería Progreso. c/General Jordana, 28. Elda. Teléfono: (965) 38 11 45.

BADAJOZ
- Control Sistemas. Avda. Santa Marina, 25A. Teléfono: (924) 25 88 00.

BARCELONA
- Catinsa Informática, c/de L'Esglesia, 15. Teléfono: (93) 784 27 17. Terrasa
- Comercial Clapera. c/Mariano Maspons, 4. Granollers. Teléfono (93) 870 45 42.
- Computer Service. Avda. Abad Marcet, 325. Tarrasa. Teléfono: (93) 788 63 77.
- Computerhard c/s. Jaime, 48. Granollers. Teléfono: (93) 870 09 19.
- Electrodomésticos Mirambell, c/Rabal, 45. San Sadurni d'Anoia. Teléf.: (93) 891 11 34
- Gadesa. Les Valls, 12-14. Sabadell. Teléfono: (93) 725 25 43
- Librería Emilia Pérez Radua. c/Mayor, 35. Castellar del Vallés. Teléfono: (93) 714 89 51.
- Librería Michel. Rda. Guinardo, 1. Sardañola.
- Mozart. c/Jaime 1, 145. Mollet. Teléfono: (93) 593 75 01.
- Radio Watt. Paseo de Gracia, 30. Teléfono: (93) 237 11 82.
- Sonimóvil. c/Alcalde Armengou, 53. Manresa. Teléfono: (93) 873 78 17.
- Suministradora Z, S.A. Av. Barbera, 49-51 Sabadell. Teléfono: (93) 710 56 66.
- Tronic. Bigay, 11-13. Tel.: (93) 212 85 96.

BILBAO
- Bilbomicro, S. A. c/Aureliano Valle, 7. 48010 Bilbao. Teléfono: (94) 443 43 51.

BURGOS
- E.I.S.A. c/Madrid, 4. Teléfono: (947) 20 46 24.

CACERES
- Informática Vivas White. San Pedro, 8. Teléfono: (927) 24 40 96.

CADIZ
- Vídeo Computer. Comandante Gómez Ortega, 59. Algeciras. Tel.: (956) 65 39 02

CASTELLON
- Horizontes. c/Navarra, 76.

CIUDAD REAL
- Electrónica Turrillo. c/Pedrera Baja, 7 Teléfono: (926) 22 38 67.

LA CORUÑA
- Cetronic, S.L. c/Palomar, 22 Bajo. Teléfono: (981) 27 26 54
- Gesty Computer. Avda. Romero Donallo, 25. Tel.: (981) 59 87 54. Santiago de Compostela
- Sanlusa, S.L. San Luis, 46 al 50. Teléfono: (981) 23 07 49.

GERONA
- Digit Informática, c/Avda. 11 Setembre, 7. Olot. Teléfono: (972) 26 94 01.
- Microchip. c/Aigua, 3. Olot. Tel.: 26 36 63.
- Regiscompte, S.A. c/Emilio Grahit, 17 Bis. Teléfono: (972) 21 99 88

GRANADA
- Rafael Moreno Torres. Recogidas, 24. Edif. Castro. Telefono: (958) 26 20 50

HUELVA
- Computerlog, S.A. c/Tendaleras, 15 Teléfono: (955) 25 81 99

IBIZA
- Kelson. Plaza España, S/N. San Antonio Abad. Teléfono: (971) 34 13 09.

LEON
- Logdata. c/Burgo Nuevo, 4 Teléfono: (987) 20 42 89
- MicroBierzo. c/Carlos I, 2 Teléfono: (987) 41 74 21 (Ponferrada).

MADRID
- Electrónica Lugo. c/Barquillo, 40.
- Key Informática. c/Embajadores, 90 Teléfono: (91) 227 09 80.
- Librería García Peña, c/Cavanilles, 52.
- Micromundo, S.A. El Zoco. Majadahonda. Teléfono: (91) 638 13 89.
- Micros Garden, c/Francisco Silvela, 19 Teléfono: (91) 401 07 27
- País de los Microordenadores. Bravo Murillo, 18. Tels.: (91) 446 33 17 - 446 37 12.
- Remshop. Galileo, 4. Teléfono: (91) 445 28 08

MALAGA
- Informática Martínez, S.A. c/Cristo de la Epidemia, 90. Teléfono: (952) 26 37 68.

MURCIA
- Procoinsa. Ronda Norte, 27. Teléfono: (968) 23 94 49.

ORENSE
- Bermello. Gral. Franco, 123.

PAMPLONA
- Itar Computers, S.A. c/Alfonso el Batallador, 16 (Trasera). Tel. (948) 27 64 04.

SALAMANCA
- Info-Gest c/Ronda del Corpus, 2-1º centro. Teléfono: (923) 21 59 93.

SAN SEBASTIAN
- Donmicro, S.A. c/Arrasate, 6. Teléfono: (943) 42 35 10.

SANTANDER
- Librería Hernández. San Francisco, 15. Teléfono: (942) 22 53 30.

SEVILLA
- Papelería Mora. c/Santa Cruz, 5. Ecija. Teléfono: (954) 83 14 80.

TARRAGONA
- Comercial Informática de Tarragona, S.A. c/Gasómetro, 20. Teléfono: (977) 23 08 53.
- Oficomplet. Plaza de la Cinta, 6. Tortosa. Teléfono: (977) 44 14 50.

VALENCIA
- Ineleksa. c/Remedio, 38-bajo-dcha. Sagunto. Teléfono: (96) 246 21 10 - 246 09 29.
- Librería Mayte. Pintor Benedito, 3. Teléfono: (96) 325 28 83

VALLADOLID
- Chips + Tips, S.A. c/Juan de Juni, 3. Teléfono: (983) 33 40 00.

ZARAGOZA
- ADA Computer. Centro Independencia. Pº Independencia, 24-26. Tel.: (976) 29 85 62.

DYNADATA
INFORMATICA

brother

BROTHER, entre los cinco mayores productores del mundo de máquinas de escribir e impresoras para la informática fue seleccionada como la marca oficial de los Juegos Olímpicos de Los Angeles 84. Con una antigüedad de 50 años como fabricante hoy día BROTHER cuenta en servicio con más de 12 millones de máquinas de escribir las cuales tienen hasta 80 tipos de diseños y teclados para 22 diferentes idiomas. Sus aproximadamente 20.000 empleados y 7 plantas mantienen una producción corriente anual de 1 millón de aparatos, junto con otros productos BROTHER.



Impresoras de margarita

HR-1:

Muy duradero y fiable, este modelo tiene un carro ancho que permite imprimir 132 caracteres en un papel de 420 mm. de ancho, ideal para grandes trabajos de oficina. Sus cassettes de cinta de imprimir son excepcionalmente económicos y permiten ahorrar hasta 7 ptas. en un folio de texto de 1.000 caracteres sobre otras impresoras de características similares. Se puede acoplar un alimentador de tractor para papel continuo o un alimentador automático de hojas sueltas.



HR-15, HR-25, HR-35:

Una familia de impresoras de alta tecnología y resolución en su sistema de impresión, tipo margarita, nos ofrecen una compatibilidad con una amplia selección de hardware existente en el mercado, en adición a la amplia selección de procesadores de datos y software. Todos los modelos tienen la cualidad de imprimir en dos colores (negro y rojo), posibilidad de remarcar textos, impresión de super y sub-índices y control de impacto para copias múltiples. Entre otras opciones se encuentran un alimentador automático de papel cortado con un máximo de 200 hojas sin parar y un alimentador de tractor que permite utilizar papel continuo. El HR-15 incorpora una memoria de tampón de 3kbytes (5 opcional) con la posibilidad de conectar un teclado para convertir el conjunto en una máquina de escribir electrónica. Imprime a una velocidad de 13 caracteres/seg. Los otros dos modelos nos ofrecen alta velocidad de impresión, 23 caracteres/seg. para el HR-25 y 32 caracteres/seg. para el HR-35. Este último modelo posee 7kbytes de memoria de tampón y ofrece 25 kbytes como opcional. Con estas memorias el ordenador al que está conectado la impresora se encuentra libre para realizar otras operaciones prácticamente en el momento en que se da el comando de "IMPRIMIR"



Impresoras de matriz

HR-5:

Imagine Ud. una impresora de que imprime gráficas y diagramas unidireccionalmente y textos bidireccionalmente en papel corriente, y que además cabe en su cartera dejando el suficiente espacio para acomodar su ordenador portátil u ordenador personal. Este ordenador es más que esto!. Ud. puede imprimir cuando y donde Ud. quiera puesto que puede operar con 4 pilas de linternas convencionales. Puede imprimir una página entera de 80 columnas de su texto sin apenas sonido. Incorpora el interface de CENTRONICS o el RS-232. Su cabeza impresora de alta calidad puede imprimir en papel normal o térmico a una velocidad de 30 caracteres/seg.

M-1009:

La impresora de matriz BROTHER M-1009 ha sido diseñada teniendo en mente la demanda tan grande que existe en fabricar una impresora de bajo costo y alta calidad accesible al bolsillo de aquellos que disfrutan de ordenadores personales en sus hogares.

Entre sus características se encuentran una cabeza impresora de matriz con una garantía de 20 millones de caracteres de vida, imprime a 50 caracteres/seg., bidireccionalmente para los caracteres corrientes y unidireccionalmente para gráficas, diagramas, y super y sub-índices. Baja en sonido, la M-1009 posee además otras funciones de alto precio y calidad como el 96 ASC II con caracteres gráficos e internacionales. Interface de Centronics o RS-232 C. La M 1009 imprime en papel cortado, con alimentadores de hojas incorporados, y con papel continuo. El M 1009 es una unidad eficaz que rendirá largos años de servicio a un costo mínimo de mantenimiento.



2024L:

El extraordinario modelo, BROTHER 2024L - ¡Dos impresoras en una!. A 80 caracteres/seg., imprime con calidad impecable para tratamiento de textos, y apretando un botón, imprime a la alta velocidad de 160 caracteres/seg. requerida para contabilidad. Esta versatilidad permite un gran ahorro en su compra y en el trabajo diario de oficina. Incluye, además, otras características sobresalientes de una gran impresora como: la posibilidad de cambiar los tipos y tamaños de caracteres entre normal, condensado, pica, élite y proporcional, la cantidad de caracteres por línea entre 136, 272 y proporcional, los espacios entre líneas entre 1/", 1/8", 14/120", n/60", n/120", la velocidad de alimentación de líneas entre 2,0"/seg. y 3,0"/seg., y puede alimentar papel con anchura entre 127 mm y 381 mm por fricción o continuo por medio de su tractor incorporado y con capacidad de imprimir un original y cuatro copias con un nivel de ruido inferior a 65 db A.

DISTRIBUIDOR
Sor Angela de la Cruz, 24. MADRID-20
Teléfs.: (91) 279 21 85 - 279 28 01 - 270 01 93 - 270 76 75.

DELEGACION: Aribau, 61, entlo. Barcelona-11
Tlf. (93) 254 65 48


commodore 16
POWER


commodore
COMPUTER